SUPER ROM

Lucid Spreadsheet Write ROM Database Outliner

NOW AVAILABLE ON THE M-200

NOW YOU CAN REALLY HAVE IT ALL!

All on one ROM. Truly the finest four programs available for the Model 100 — guaranteed. Try it for 30 days. If you aren't blown away by the excellence return it for a full refund.

$199.95

The four best programs for the Model 100 all on one ROM. 32K of power without using any RAM for program storage. This is the PCSG Snap-In ROM that just presses easily into the little ROM socket in the compartment on the back. You access the four right from the main menu like built-ins.

Write ROM — the definitive word processor for the Model 100. Function key formatting or dot commands. Search and replace. Library feature — inserts words, phrases or whole documents into text from just a code. MAP lets you see a picture of your document. In all there are 60 features and functions. No one can claim faster operation. FORM lets you create interactive forms with on-screen prompts that you can answer from the keyboard. Nothing else for the Model 100 compares with the features of Write ROM. Exactly the same as the Write ROM sold as a single program. Infoworld says it "makes the Model 100 a viable writing unit ... surpassed our highest expectations for quality and clarity."

Lucid Spreadsheet: This is the one PICO magazine says "blows Multiplan right out of the socket" and Infoworld performance rated as "excellent" and said "makes the Model 100 compute." Gives you features you cannot get with Lotus 123. Lets you build spreadsheets in your Model 100 that would consume 140-150K on a desktop. Program generating capability with no programming knowledge required. Variable column widths. Includes find and sort with function key control. It's fast, recalculates like lightning. No feature has been taken from the original, only new ones added.

Database: This is a relational data base like no other. You can do everything from mailing lists to invoices. No complicated pseudo-coding, you create input screens as simply as typing into TEXT. You are not limited by size; you can have as large an input screen as you wish. Prints out reports or forms, getting information from as many files as you like. Complete math between fields. Total interface with Lucid worksheets.

Outliner: Does everything that Thinktank does on a PC but a whole lot better. Includes a Sort for your headlines. Lets you have headlines of up to 240 characters. Has cloning, hoisting and sideways scroll up to 250 characters. Like Lucid, this one sets a new standard for outliners. This is the way to plan and organize your projects.

Present Lucid and Write ROM owners can upgrade for $100. If you have both it's $75.

As usual PCSG sells the Super ROM on a thirty day guarantee. If for any reason you are not satisfied, simply return it for a full refund.

We are excited about this product. Super ROM gives the Model 100 the true power of a desktop. No other multi-program ROM has software that compares. But don't take our word for it. We invite you to make that comparison yourself. Priced at $199.95 on Snap-In ROM.

Got stuck with somebody else's multi-ROM? We'll upgrade it for $100.

(214) 351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd., #206, Dallas, TX 75229

MC, Visa, American Express, Check, or C.O.D.

For Tandy's 100 200 and 600 Portables and 1000, 1200, 2000 and 3000 MS-DOS Computers

PCM

The Personal Computer Magazine for Tandy® Computer Users

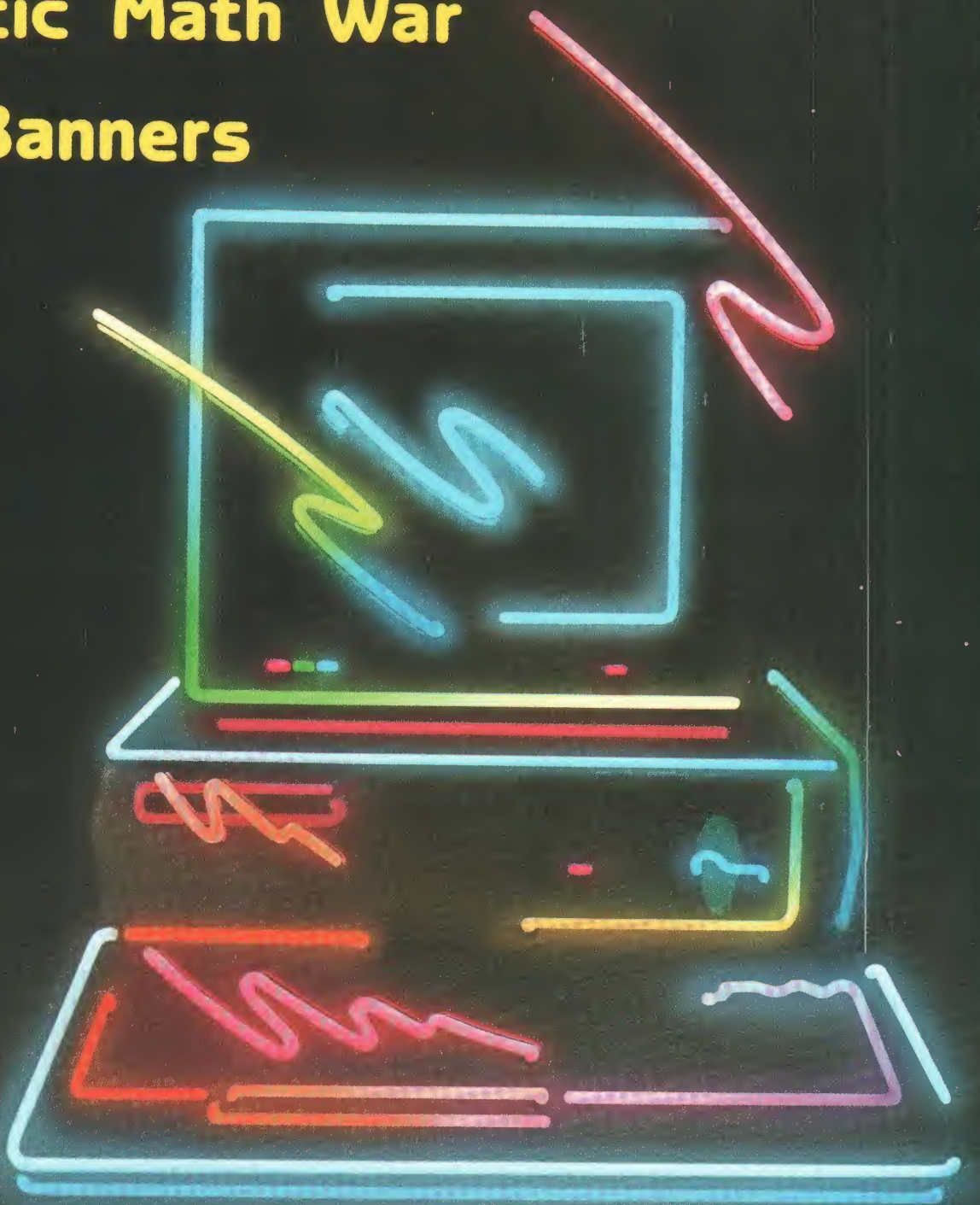
Vol. III No. 10
April 1986

U.S. $3
Canada $3.95

BASIC for Beginners

Galactic Math War

Bold Banners



F W Crawford ©1986

6 ROM BANK

plus powerful Nicad battery pack that gives 30 hours of power to your Model 100

$269

A joint PCSG/CRYPTRONICS project.

Now you can access LUCID, DISK+, WRITE ROM, RAM+ and others instantly.

The 6 ROM BANK plus battery pack lets you have the ultimate Model 100 system. When you get your unit just press in up to 6 ROMS into the sockets then push in a simple plug into your Model 100's ROM socket. The ROM bank has two posts on either end that insert into the two little holes on the underside of your Model 100.

The ROM bank props up the Model 100 at the same angle and height as those little legs you've seen. The ROM BANK itself is only about 1½" deep and it runs the width of your Model 100. It only weighs one pound. It not only installs instantly, but it pops free in a second if you need everything to lie flat in a briefcase.

Change from ROM to ROM with the touch of a thumb switch.

You can go from LUCID to WRITE to DISK+ to any other ROMS just by turning the thumb switch at the side of the ROM bank. The 6 ROM BANK is a sturdy well built construction that looks like it is a part of your Model 100.

What is also fantastic is that the ROM bank has a powerful NICAD battery and recharger built right in. This power source supplies up to 30 hours of life to your Model 100 with just 6 hours of recharge. What's nice is that it recharges right from your Model 100's power adapter. This is a quick charge system and if you need power in a hurry, you can get 6 hours of life for your Model 100 by just charging for an hour and half.

Suddenly, the Model 100 is a very powerful computer.

If you have the HOLMES portable disk drive with its powerful bundled database software package or the CRYPTRONICS 128K RAM expansion along with LUCID, WRITE ROM and DISK+ in the 6 ROM bank with its rapid rechargable NICAD power source, you have the ultimate portable system. It's all available only from PCSG.

As usual we don't want you to take our word for it. The 6 ROM BANK is sold on a 30 day trial. If you are not satisfied simply return within thirty days for a full refund. MC, VISA, AM. EX. or C.O.D.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

VISA / MASTER CARD / AM. EXP. / C.O.D.

NOW AVAILABLE ON THE M-200

PCM IS NOW ON DISK

Each and every month, PCM offers an assortment of programs for the home and office — utilities, business applications, games and more — all for the price of typing in the listings.

If you're a person on the move, and too busy to spend time typing lengthy listings, then PCM has just what you need — **PCM ON DISK.**

PCM ON DISK provides each issue's MS-DOS programs ready to load and run with only a few keystrokes!

AS AN ADDED BONUS, **PCM ON DISK** will include PCM's portable computer programs, which may be downloaded to portables for those readers who have both an MS-DOS machine and one of Tandy's portable computers.

$99 for a year's subscription to PCM ON DISK!

That means you get a disk each month for a year, containing all programs listed in that month's issue, for an average cost of only $8.25!* We think it's worth your time to subscribe.

*Disks for individual issues (beginning this month) may be ordered for only $12 each.

PCM

PCM ON DISK
The Falsoft Building
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

☐ Full Year's Subscription (specify starting month) _____

☐ Single Volume (specify month(s) & year) _____

Name _____

Address _____

City _____ State_____ ZIP _____

Payment Enclosed ☐

Charge Visa ☐ MasterCard ☐ American Express ☐

My Account# _____ Card Exp. Date _____

Signature _____

*Subscriptions to PCM ON DISK are $99 a year in the United States. Canadian rate U.S. $115. All other countries U.S. $130; PCM ON DISK single copy rate is: $12 within the U.S.; U.S. $14, Canadian rate; U.S. $16, all other countries. Kentucky residents add 5% sales tax. All subscriptions begin with the current issue unless otherwise specified. Please allow 5 to 6 weeks for first copy. **U.S. currency *only*, please**. In order to hold down non-editorial costs, we do not bill.

PCM

The Personal Computer Magazine for Tandy® Computer Users

April 1986 Vol. III No. 10

Editor and Publisher
Lawrence C. Falk

Managing Editor Danny Humphress
Senior Editor Tamara Renee Dunn
Submissions Editor Jutta Kapfhammer
Reviews Editor E. Monica Dorth
Editorial Assistants Jo Anna Arnott, Wendy Falk, Judi Hutchinson, Angela Kapfhammer, Shirley Morgan
Technical Assistants Cray Augsburg, Chris Wehner
Contributing Editors William Barden, Jr., Bob Covington, John B. Harrell III
Consulting Editors Ed Ellers, Belinda C. Kirby, T. Kevin Nickols

Art Director Tracey Jones
Designers Jody Gilbert, Heidi Maxedon, Kevin Quiggins, Sandra Underwood
Production Assistant Cindy Jett

Chief of Typography Debbie Hartley
Typography Services Jody Doyle, Suzanne Benish Kurowsky

Falsoft, Inc.

President Lawrence C. Falk

General Manager Patricia H. Hirsch
Asst. General Mgr. for Finance Donna Shuck
Admin. Asst. to the Publisher Sue E. Rodgers
PCMfest Coordinator Judy Brashear

Editorial Director James E. Reed
Asst. Editorial Director Jutta Kapfhammer
Creative Director Jerry McKiernan
Manager of Public Relations Holly Weaver

Chief Bookkeeper Diane Moore
Advertising Accounts Beverly Taylor
Dealer Accounts Judy Quashnock
Accounts Payable/ADP Lisa Ragan
Director of Fulfillment Services Bonnie Frowenfeld
Asst. Dir. of Fulfillment Services Sandy Apple
Asst. Customer Service Manager Beverly Bearden
Word Processor Manager Patricia Eaton
Chief of Printing Services Melba Smith
Pre-press Production John Pike
Dispatch Janice Eastburn
Asst. Dispatch Mark Herndon
Business Assistants Laurie Falk, Doug Orr, Sharon Smith, Monica Wheat, Pam Workhoven

Advertising Coordinator Doris Taylor
Advertising Representative Kim Vincent
Advertising Assistant Debbie Baxter
(502) 228-4492

West Coast Advertising and Marketing Office
Director Cindy Shackleford
Advertising Representative Shirley Duranseau

For PCM Advertising and Marketing Office Information, see Page 130

PCM — The Personal Computer Magazine for Tandy® Computer Users is published every month of the year by FALSOFT, INC., The Falsoft Building, P.O. Box 385, Prospect, KY, 40059. Phone (502) 228-4492. **PCM — The Personal Computer Magazine for Tandy® Computer Users** and the **PCM** logotypes are registered ® trademarks of FALSOFT, Inc.

Second class postage paid Prospect, KY, and additional offices. USPS N 713-470 (ISSN 0747-0460). POSTMASTER: Send address changes to PCM, P.O. Box 385, Prospect, KY 40059. Forwarding Postage Guaranteed.

Subscriptions to **PCM — The Personal Computer Magazine for Tandy® Computer Users** are $28 per year in the United States. Canadian rates are U.S. $35. Surface mail to other countries is U.S. $64, air mail U.S. $85. All subscriptions begin with the next available issue.

Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in United States currency only. Full refund after mailing of one issue. After two issues have been mailed, 10/12ths of the subscription price will be refunded. NO refund after mailing of three or more magazines.

Limited back issues are available. Please see notice for issues that are in print and their costs. Payment accepted by VISA, MasterCard, American Express, Cash, Check or Money Order in U.S. currency only.

Contents

Features

Departments

Reviews

Cover illustration copyright© 1986 by Fred Crawford

Tandy computers offer everything ...almost

Check out the software that completes their systems —*filePro*®

Tandy MS-DOS Software Comparison Chart

	pfs® File/Report	dBase™ II	*filePro* 16 Smallware
GENERAL CHARACTERISTICS:			
Menu driven	yes	no	YES ✓
Allows user to create integrated business systems	no	programmer required	YES ✓
Developed systems and data can be moved to multi-user environments	no	no	YES ✓
Professional support available from the software's authors	no	no	YES ✓
PRICE	$265	$595	$495
CAPACITIES:			
Fields per record	100	32	999 ✓
Characters per record	1679	1000	4608 ✓
Records per file	1300	65535	16,000,000 ✓
Indexes per file	1	7	12 ✓
Number of digits per numeric field	20	10	24 ✓
Number of files usable concurrently	1	2	10 ✓
Files span multiple drives	no	no	up to 8 ✓
FEATURES:			
Full-screen facility for creating custom screen layouts	yes	no	YES ✓
Full-screen facility for creating custom report layouts	no	no	YES ✓
Built-in field types (error checking)	no	3	12 ✓
User-defined field types	no	programmer required	200 ✓
Conditional math	no	programmer required	YES ✓
User-defined menus	no	programmer required	YES ✓
Change file layout without losing existing data	possible	possible	automatic ✓
Data protection	no	programmer required	YES ✓
Password security	no	programmer required	YES ✓

The new Tandy computers can really speed things up. If you don't let obsolete software bog you down.

Try *filePro* 16, the latest generation of data base software from the authors of Profile®.

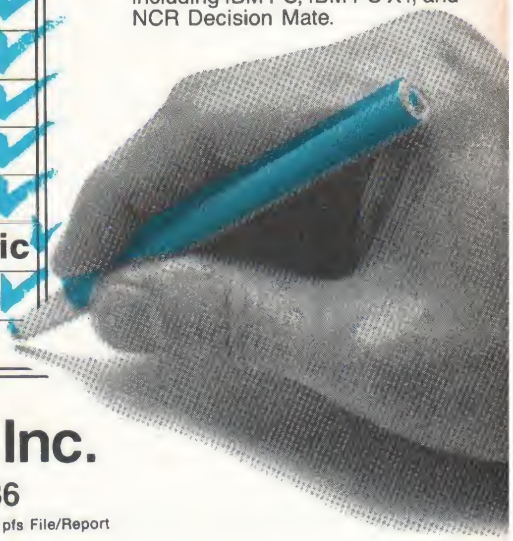
Choose it for its power (check our capacities); its ease-of-use (check our features); and its unique ability to grow as your needs grow, and even move the systems you create to multi-user computers.

filePro is available for the Tandy 1000, 1200 and 2000 computers (hard disk required). Available directly from us or through your Radio Shack store's Express Order Software service.

To order, call (800) 847-4740.

For customer service or when dialing in New York State, call **(212) 398-9290.** Ask for Smallware sales. Visa and MasterCard accepted.

Also available for other computers, including IBM PC, IBM PC-XT, and NCR Decision Mate.



The Small Computer Company, Inc.

230 West 41st Street, Suite 1200, New York, New York 10036

✓240

Smallware and *filePro* are registered trademarks of The Small Computer Company, Inc. dBase II is a trademark of Ashton-Tate. pfs File/Report is a registered trademark of Software Publishing. Profile is a registered trademark of Tandy Corporation.

Exciting Times

These are heady and exciting times here at PCM. Back in the early days of our publishing history, I was the one who opened the mail every day because, primarily, I was the only one working here. Over the years, we've added a few people and someone else opens the mail every weekday. Some days, even with some machines to help do the job, it takes a couple of hours.

But on Saturday mornings, I still get the mail and go through it (they won't let me *open* it because they're afraid I'll mess something up). It's something I really like to do — it keeps me "current" on what is coming in and what is happening here.

For the past couple of months, it's been more exciting than it has been for a long time. And much of the excitement revolves around PCM. We're getting more mail, more letters, more subscription orders and the like than ever before.

Of course, THE RAINBOW, our first magazine, still gets tons of mail. But, over the years, we've grown accustomed to that volume.

Someone asked me the other day whether Tandy "really did" sell more 1000s over the last few months than IBM sold PCs. Well, based on our mail, I'd have to certainly agree! The mail volume seems to be an upward curve that grows exponentially.

You're holding one result of that in your hands — a larger PCM. Size of magazines is based primarily on the amount of advertising space that we sell, and as this market continues to grow by leaps and bounds, so does the advertising. As it grows, so does the size of the magazine. As you can see, we're up to 132 pages this month. Less than a year ago we were at 78. And, while that means more pages of advertising income, it also means more pages of programs, articles, tutorials and reviews for you to read and use. It works for all of us.

Another result of this increase in our ranks is that we're getting more and more calls from software and hardware manufacturers seeking entry into this burgeoning market. Often, the question is, "Will it run on the 1000?" or the 2000 or the 100 or the 3000 or the 600. We're willing to make these tests for them. It is to their advantage, ours and, naturally, yours. The larger the software and hardware base, the more useful your Tandy MS-DOS or portable is.

What is the most interesting of all is the willingness of many manufacturers to make modifications to their programs, if necessary, to "get into the Tandy market." That contrasts sharply with the don't-bother-me attitude as recently as a year ago from many people.

Yes, these are exciting times for us all — for you, as a user — for me, as a publisher and enthusiastic supporter of this market. And, of course, for the people at Tandy Center as well.

We'd like to keep on increasing the size and usefulness of PCM and you can help us. When you call or write for information or to place an order with one of our advertisers, please mention that you "saw it in PCM." I know that is a trite expression — but you would be very surprised at how much good it can do. And for those from whom you buy things that *don't* advertise here, please let them know you read PCM. After all, you are supporting them by buying their products — ask them to support your market by being involved in it. You'd be surprised how much *that* helps, too.

We have something really exciting happening here. Let's keep it going! After all, it is in all of our interests.

— Lonnie Falk

Problems in the 'Print Shop'

Editor:

I recently purchased the program *Print Shop*, by Broderbund Software, to run on my Tandy 1000. I have an Epson MX-80 printer with Graftrac Plus. My problem is that I can't print the graphics I create. The problem allows me to either print the printer test or print the first graphics created, then if I try to print a second time it tells me the printer is not ready. I called Broderbund and the person I talked with said they had other complaints from people who had a Tandy 1000 and an Epson printer. The only suggestion she could give me was to turn off the computer after each printout and start over again.

There has to be a better way. I also have the program *Print Master*, by Unison World Inc., and that program works fine although I would like to hear from anyone who has been able to move *Print Master* to hard disk successfully. I understand there is a utility that will do this, but I have not been able to find it.

I would appreciate hearing from anyone who has found the fix for the *Print Shop* problem or anyone who knows how to move *Print Master* to hard disk.

Larry E. Ierley
Elizabethtown, PA

Editor:

I just finished reading an article in your February 1986 copy of PCM entitled "Upgrade Your Tandy 1000," by Brian Alsop. In the article Mr. Alsop states that he used a PBJ MFB-1000 to upgrade his Tandy 1000. I, too, have the PBJ MFB-1000 installed in my Tandy 1000 along with a 300 Baud Modem Board, Cat. No. 25-1003. In his statement about the drawbacks to the PBJ concerting the COM1 and COM2 ports, he says you have to option your other communication expansion boards to COM2 and use the MFB as COM1 only. However, you can option the PBJ board as COM2, as the user's manual shows on Page 5, Section 4 of the MFB-1000 user's manual. It shows you how to reconfigure the MFB to COM2 by moving two jumpers, or plugs, on the board. The first plug is labeled COM1/COM2 and is located near U23, to the left of center of the board, and the next plug is labeled E1/E2, and is located immediately to the left of the edge connector. All of this information is detailed on pages 5 and 6 of the user's manual.

I have been using my MFB with 512K and have had no trouble with the operation of the board or my system. I recommend this board to anyone who is thinking of expanding their system.

David Baker
Essex, MD

Editor's Note: There is no mention in the owner's manual of the MFB-1000 board we received for review of a way to configure the board as COM2. We did, however, receive a number of letters informing us otherwise.

A Word about *Word*

Editor:

On Page 49 of the Microsoft Press publication, *Word*, by Janet Rampa, is a caption titled "Scrolling Sideways." With my T2K, which doesn't have a key labeled scroll lock, the technique of scrolling sideways won't work or doesn't work for its operator.

Does this mean either *Word* a la IBM is greater than *Word* a la T2K, or one is greater, capability wise, than the other regardless of whichever IBM model — PC, PC*jr*, XT or AT — is being compared?

Also, is there a print wheel available for Tandy's DWP-210 that has the symbol used in the lawyer world to denote a section?

H.E. Lamb
Camp Hill, PA

Reply from Tandy Corporation

Dear Mr. Lamb:

I'm writing in response to your January 16 letter to PCM magazine.

When using the Microsoft *Word* program on the Tandy 2000, sideways scrolling can be achieved by using the corresponding left- or right-arrow keys. Text must be formatted wider than the display screen for the sideways scrolling to occur.

Our in-house software experts tell me that Tandy's *Word* program, Version 1.0, is comparable to Microsoft's *Word* Version 1.0. Microsoft has come out with more recent versions of the program with additional capabilities.

We do not carry a daisy wheel for the DWP-210 printer that has the symbol used by lawyers to denote a section.

I hope this answers your questions. Please let me know if you need more information.

Amy Arutt
Assistant Manager
Marketing Information
Tandy Corporation
Ft. Worth, TX

Tandy 1000 Lover

Editor:

I am a very happy owner of a Tandy 1000 computer with a Tandy CM-4 color monitor and an Epson RX-80 printer. The Tandy 1000 computer has better graphics than any other computer [I've seen]. I love the graphics mode of 320 by 200 in 16 colors (Screen 5) very much because I can draw pictures in 16 colors instead of two or four colors as on any other computer. I can use the PAINT(X,Y) ,A$,C command to paint the pictures in 256 colors.

I love the Microsoft BASIC that comes with the computer: It is more powerful than BASICs on any other computers because it has more BASIC commands. The BASIC commands for graphics are more powerful, beating the Extended Color BASIC in the Radio Shack Color Computer. After playing with the Radio Shack Model 1 computer, Radio Shack Color Computer, Atari 1200XL computer and Commodore 64 computer, I fell in love with my Tandy 1000 because it is the most powerful computer I have owned. Too bad the Tandy 1000 computer does not have sprites like the Commodore 64 and 128 computers have, but I can simulate sprites with GET(X1,Y1)-(X2,Y2),A% and PUT(X,Y),A%,XOR commands.

My mother bought me the Tandy 1000 computer with one disk drive and 128K RAM. I would like to add more memory, a clock/calendar and serial ports to my computer. I could buy three expansion boards from Radio Shack, wasting $440, or buy one board with these options cheaper from another company. I am not sure which expansion board to buy. Please tell me which board to buy. I need a second disk drive so I don't have to change disks all the time when copying programs from one disk to another. I will buy the disk drive from Radio Shack. Should I have Radio Shack install the disk drive? Is it too hard to install the disk drive myself? I can squeeze 12K or 14K out of my computer by moving the text screen to the top of memory and using CLEAR ,,,XXXX to get more memory. I found this from *80 Micro* magazine. Here is the program that does this:

```
10 SCREEN 0,1,0,0
20 WIDTH 40
30 SCREEN 0,1,7,7
40 CLEAR ,,,2048

10 SCREEN 0,1,0,0
20 WIDTH 80
30 SCREEN 0,1,3,3
40 CLEAR ,,,4096
```

Jeffrey Morris
Rochester, NY

Editor's Note: There are at least two manufacturers of multifunction boards for the Tandy 1000: PBJ's MFB-1000 and Hard Drive Specialist's TanPak. In addition to these two, Tandy now sells the MemoryPlus expansion board which can expand your machine to 640K and allows you to choose a clock/calendar/mouse or RS-232 port option.

It is not difficult to install a second floppy disk drive in your computer yourself. We do recommend, however, that you purchase your disk drive from a company that advertises support for the Tandy 1000 — the instructions should be easier to follow for your machine.

DISK POWER

. . . AT LAST !!

*ULTRA*CHARGE YOUR TANDY PORTABLE DISK DRIVE WITH DISK POWER 100/200

- **A REAL DISK OPERATING SYSTEM**

All disk functions completely supported by **DISK POWER**. No need to rely on FLOPPY.CO or user-written programs for external utilities. **DISK POWER** is not just another disk manager program, it is the full featured DOS that the MODEL 100 should have had from the very beginning!

- **DISK ACCESSIBLE FROM BASIC & TEXT**

Data file I/O operations supported with familiar BASIC commands. Save and Load programs from within BASIC. From TEXT, every .DO file can be Loaded from or Saved to Disk by **DISK POWER**.

- **TELCOM ENHANCEMENTS**

Direct Access to Basic & TEXT from inside TELCOM. Kill, Load, Save or Edit files without losing telephone communications. You can see your files while uploading. With **DISK POWER**, it is all possible!

- **FULLY MENU DRIVEN**

Any DISK or RAM file is selected by the cursor. The stroke of a SINGLE function key will perform the desired task on the HIGHLIGHTED file. You may never have to remember or even type a filename again, when using **DISK POWER!**

- **NOT JUST "FOR HACKERS ONLY"**

Though programmers will be surprised by the versatility of **DISK POWER**, regular users will find our operating system FAST, easy to use AND easy to understand.

- **WRITTEN BY HUGO FERREYRA** (SoftPower)

In **DISK POWER**, you can expect to find the same ***ULTRA***-features, innovation & ease of use that all users & reviewers found in **TEXT POWER 100**, which made it **"THE BEST FORMATTER THERE IS FOR THE 100"**.

- **24 HR/DAY BBS TECHNICAL INFORMATION & SUPPORT HOTLINE: (514) 457-2917**

Use STAT **M8N1E**, and Log-on with ID: **DISK** using PASSWORD: **POWER**

DISK POWER 100/200 **$49.95** US **TEXT POWER 100** **$49.95** US

or SAVE by buying the ***ULTRA* POWER** program pack: DISK POWER/TEXT POWER 100 FOR ONLY **$89.95** US

To order *TEXT POWER 100*, specify printer(s) used. The basic price includes support for one printer. Please add $10.00 for each additional printer you wish supported. All U.S. orders shipped from within the USA. Absolutely no custom forms, sales tax or duty required. Add $3.00 S&H to each order.

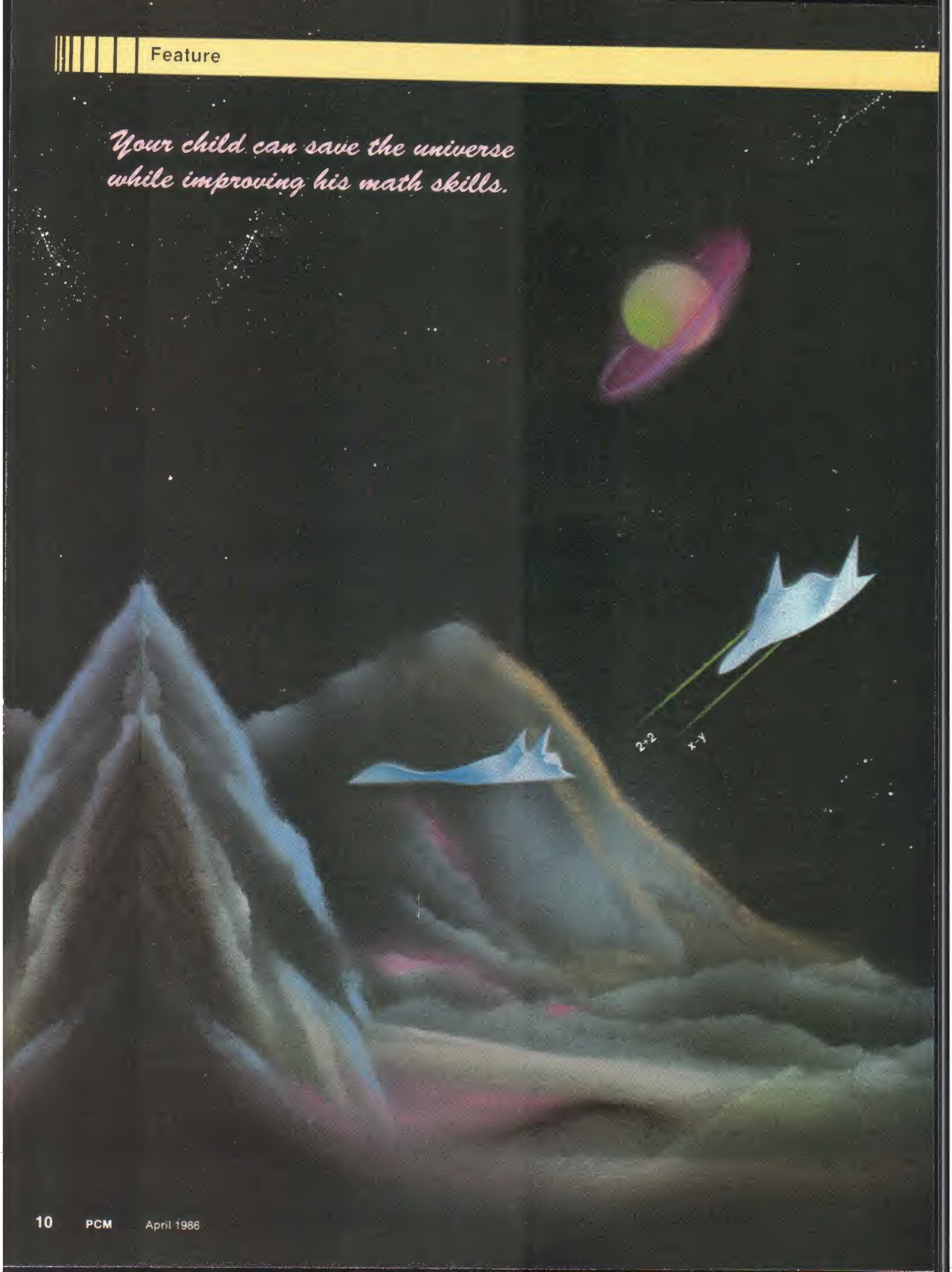
No surcharge for VISA or MC
To order, call **TOLL FREE:**

USA (Except N.Y.)	**1-800-448-4511 Ext. 313**
NEW YORK STATE ONLY	**1-800-962-1380 Ext. 313**
ONTARIO & QUEBEC	**1-800-255-1133 Ext. 313**

or send CHECK or MONEY ORDER to:

ULTRA SOFT P.O. Box 219, Ste Anne de Bellevue, Québec
CANADA H9X 3R9 (514) 457-9293 (9-5 EST)

Your child can save the universe while improving his math skills.



Galactic Math War

By Robert C. Mills

Disguised as a game, this entertaining math tutor grows with the skill of the player. When saving the galaxy becomes too easy or too difficult to be fun anymore, simply raise or lower the difficulty level. Let's face it, an educational program that is too easy or too hard just doesn't teach what it is designed to and is soon discarded. The configuration program for this game allows anyone (even the player) to set the game's level of difficulty.

The player is the pilot of an advanced space shuttle with the mission of saving the Galaxy from the evil invaders. By entering the correct answer to addition or subtraction problems, the pilot's lasers drain all the energy from the enemy shuttle. If the pilot enters an incorrect answer, the enemy drains away one of the shuttle's shields. When all three shields are gone the shuttle is vaporized. To save the Galaxy, the pilot must drain the energy from five enemy shuttles in each of five difficulty levels.

Bob Mills is a master sergeant in the United States Air Force stationed at the Armed Forces Institute of Pathology in Washington, D.C. He is the course supervisor of the Tri-Service School of Histopathology and has written numerous laboratory utility and educational programs for several USAF laboratories.

The first program, *Configure Math War*, creates or changes the data file, which contains the maximum values that can be generated by the game's random number generator. In Line 120 an attempt is made to read an existing file; if the file does not exist one is written to the disk later. After an instruction display, the existing values are displayed (lines 250-330) and the user is allowed to accept or change these values. Note that one subroutine is used (lines 850 and 860) to control all of the user friendly one-key selections. If the user decides to change the values, a flag is set (Line 360) to force the program to save the changes on the disk.

Random access data files can only store string values, but the program requires integers to operate; `LSET`, `MKI$` and `CVI` commands are used to make the conversions (see lines 910 and 980). The subroutine at lines 950-1010 write the data to the disk in the `CONFIG.WAR` data file. The handy menu at lines 730-830 lets the user restart configuration,

run *Galactic Math War* or exit to system.

The second program, *Galactic Math War*, uses the power of Tandy 1000 BASIC graphics and audio commands to create a pleasing audio/visual display. Memory is cleared to 32768 in Line 100 to allow for use of SCREEN 5. Because the use of some colors on a monochrome monitor causes an unpleasant faded look, all colors except black and gray are set to white in lines 170-200 (if color is set to 'N' during configuration).

To speed things up a bit, GET/Graphics is used to store the enemy shuttle and laser explosion pictures in memory arrays. In lines 1200 and 1240, GET stores the figures and in lines 270, 300, 430, 540 and 590 PUT/Graphics quickly places the images on the screen. In lines 140-160, the program checks the value of A(1) and COLSEL$ to be sure the program has been configured. In lines 1010-1110, the random numbers are generated and the correct answer is calculated. Note the use of the SWAP statement to ensure positive answers to subtraction problems.

This unique tutor should serve you well. I have included a few extra REM (') statements in brackets to make following the programs a little easier. You don't have to type in the extra statements (unless you just like to type). If you don't care to type at all, I will be glad to send you the finished programs on a 5¼-inch disk for $5 to cover the cost of the disk and postage. I will be happy to answer any questions you may have and would like to hear your comments. My address is 300 Heather Ridge Dr., Frederick, MD 21701; phone (301) 695-5228. □

Listing 1:

```
10 '*************************************************************************
20 '*                      GALACTIC MATH WAR                                *
30 '*            A MATH TUTOR GAME FOR THE TANDY 1000                       *
40 '*                        ROBERT C. MILLS                                *
50 '*                    300 HEATHER RIDGE DRIVE                            *
60 '*                      FREDERICK, MD. 21701                             *
70 '*                         (301) 695-5228                                *
80 '*************************************************************************
90 ' ### INITIALIZE ###
100 KEY OFF:CLEAR,,,32768!:SCREEN 5,1:DEFINT A-Z:SHIELD=3
110 RANDOMIZE VAL(RIGHT$(TIME$,2))  '[SEED NUMBER GENERATOR]
120 DIM A$(5),B$(5),A(5),B(5),C(16)
130 GOSUB 920  '  [GO GET VALUES FROM THE DATA FILE FOR RANDOM #s]
140 ' ### ERROR TRAP TO CHECK FOR DATA FILE ###
150 IF A(1)<1 THEN 1360
160 IF COLSEL$<>"Y" AND COLSEL$<>"N" THEN 1360
170 ' ### SET COLORS ON VALUE Y OR N FROM DATA FILE ###
180 IF COLSEL$="Y" THEN 190 ELSE 200
190 FOR X=0 TO 15:C(X)=X:NEXT X:GOTO 210
200 FOR X=1 TO 15:C(X)=15:NEXT X:C(0)=0:C(11)=7
210 '### INTRODUCTION PAGE AND PUT GRAPHICS IN MEMORY ###
220 ON ERROR GOTO 0  ' [INACTIVATE ERROR TRAP ROUTINE]
230 GOSUB 1120   '[DRAW AND STORE SPACE SHUTTLE]
240 SOUND ON:PLAY"MBV12O4C2G4F8E8D8O5C4O4G2"
250 COLOR 15,0
260 LOCATE 10,9:PRINT"GALACTIC MATH WAR"
270 PUT (100,100),S
280 LOCATE 19,13:PRINT"by: BOB MILLS"
290 PRINT:PRINT"     PRESS ANY KEY > "
300 A$=INKEY$:PUT (100,100),S:IF A$="" THEN 300
310 ' ### USER SELECTS ADDITION OR SUBTRACTION ###
320 CLS:LOCATE 10,5:PRINT" CHOOSE YOUR LASER STYLE: "
330 LOCATE 12,10:PRINT"1 = ADD   +":LOCATE 14,10:PRINT"2 = SUBTRACT  -"
340 A$=INKEY$:IF A$="" THEN 340
350 IF A$="1" THEN OP$="add":LET SG$="+":GOTO 390
360 IF A$="2" THEN OP$="sub":LET SG$="-":GOTO 390
370 BEEP:GOTO 320
380 ' ### START OF LOOP FOR THE 5 GAME LEVELS ###
390 FOR L=1 TO 5
400 ' ### START OF LOOP FOR THE 5 ENEMY SHUTTLES ###
410 FOR ENEMY = 5 TO 1 STEP -1
```

HARD DRIVE PRICE BREAKTHROUGH

STARTING AT $299.95!

5 to 60 megabytes complete easy to install internal and external systems
— will boot just like an XT — fully warranteed —
same day shipping from stock — fast service — free technical assistance

REMOVABLE CARTRIDGE NOW AVAILABLE
$499.95 to $999.95

STREAMING OR FLOPPY TAPE BACK UP SYSTEMS

complete internal and external systems

$1,795.00 — TURBO M — DUAL SPEED

20 megabytes — one floppy — 640K — clock calendar — serial port — parallel port
monochrome card — monochrome monitor with tilt and swivel base
Keytronics 5151 style keyboard — one year warranty — shipped from stock

MEGADISK™ HARD DISK DRIVE SYSTEMS — **TOLL FREE ORDERING 1-800-343-8841**

For the { **IBM/PC**, Tandy 1000, **TRS/80 Models I/III/IV/4P**, Compaq, PC Workalikes, **Color Computers**, Heath/Zenith, Max/80
Complete with Hardware, Cables, Software and Quikfit Installation

5 Megabytes Internal Mount	starting at	$ 239.95
10 Megabytes Internal Mount	starting at	419.95
20 Megabytes Internal Mount	starting at	519.95
5 Megabytes External System	starting at	499.95
10 Megabytes External System	starting at	639.95
20 Megabytes External System	starting at	799.95
Tape Backup System — Internal Or External	starting at	449.95

Wow! New Low Prices

DOS Systems Available: { IBM/Heath — DOS, 1.0, 2.0, 2.1, 3.0, or later
TRS/80-LDOS, TRSDOS 6.x, Newdos/80, Dosplus, CP/M, COCO DOS, Max/80 LDOS, OS9

FULLY WARRANTIED — PARTS AND LABOR — 24 HOUR SERVICE — CALL TOLL FREE — 1-800-343-8841

IBM ENHANCEMENT CARDS

Multifunction with 384K Serial port, parallel port Clock calendar, game port	**$199.95**	Color graphics	$139.95
Serial port	69.95	Monochrome/RGB Color	229.95
Monochrome only	119.95	Disk controller with cable — up to 4 drives	79.95
Monochrome graphics w/parallel port	**139.95**	384K memory	129.95
Network card	249.95	Hard drive controller	159.95
		Keytronics 5151 style keyboard	129.95

DEALER INQUIRIES INVITED
CALL 1-617-872-9090

PRICES CHANGE EVERY DAY
CALL 1-800-343-8841

PC © 1985 **SOFTWARE SUPPORT, INC.**

1 Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425

Hours: Mon. thru Fri. 9:30 am to 5:30 pm (E.S.T.) Sat. 10 am to 3:30 pm

SERVICE POLICY — Our Professional Technical Staff Is Available To Assist You Monday Through Saturday.
WARRANTIES — Up To One Full Year Parts And Labor. Floppy Disk Drive Power Supplies — Five (5) Years.
SERVICE — 24 Hour Turn-A-Round On All In-Stock Parts. Dealer Inquiries Invited. Call 617-872-9090

Please Call For Shipping, Handling And Insurance.
Cash Discounted Prices

Toll Free 1-800-343-8841

Please Call For Our Latest Price Saving Specials.

Not Responsible for Typographical Errors. Prices and Specifications May Change Without Notice.

```
42Ø GOSUB 126Ø    ' [DRAW COCKPIT]
43Ø PUT (12Ø,3Ø),S
44Ø GOSUB 1ØØØ   ' [GO GET RANDOM NUMBERS AND ANSWER TO PROBLEM]
45Ø LOCATE 19,13:PRINT N1:LOCATE 19,19:PRINT N2
46Ø LINE (98,135)-(128,16Ø),C(2),B:LINE (148,135)-(178,16Ø),C(2),B
47Ø LOCATE 19,5:PRINT SHIELD;
48Ø LOCATE 2,3:PRINT"LEVEL=";L;"                ENEMY SHIPS=";ENEMY;
49Ø LOCATE 19,28:INPUT "",G   ' [GET USER'S ANSWER]
5ØØ IF G=CORRECT THEN 52Ø ELSE 57Ø   ' [CHECK USER'S ANSWER]
51Ø ' ### CORRECT ANSWER ENTERED ###
52Ø SHIELD=SHIELD +1
53Ø LINE (5,11Ø)-(16Ø,5Ø),C(14):LINE (314,11Ø)-(16Ø,5Ø),C(14)
54Ø PUT (12Ø,3Ø),H:NOISE 4,12,18:PLAY"MFMSO5V1ØC8F8C8"
55Ø FOR Z=1 TO 5ØØ:NEXT Z
56Ø GOTO 65Ø
57Ø '### INCORRECT ANSWER ENTERED ###
58Ø LINE (5,11Ø)-(6Ø,4Ø),C(14):LINE (314,11Ø)-(6Ø,4Ø),C(14)
59Ø PUT (2Ø,28),H
6ØØ PLAY"MFO1V1ØF8"
61Ø FOR Z=1 TO 5ØØ:NEXT Z
62Ø COLOR C(2),C(4):NOISE 5,15,1Ø:FOR Z=1 TO 1ØØ:NEXT Z:COLOR 15,Ø
63Ø SHIELD=SHIELD-1:IF SHIELD=Ø THEN 82Ø   ' [CHECK IF GAME OVER]
64Ø GOTO 42Ø
65Ø NEXT ENEMY
66Ø IF L=5 THEN 72Ø   ' [CHECK IF GAME HAS BEEN WON]
67Ø CLS:LOCATE 12,15:PRINT" NEW LEVEL "
68Ø FOR Z=1 TO 25Ø:NEXT Z
69Ø PLAY"MFV1ØO4C2G4F8E8D8O5C4O4G2"
7ØØ NEXT L
71Ø ' ### WON THE GAME ###
72Ø CLS
73Ø LOCATE 12,2:PRINT" YOU HAVE SAVED THE GALAXY "
74Ø COLOR 15,Ø:BEEP:COLOR C(4), C(2):BEEP:COLOR 15,Ø:BEEP:COLOR C(4), C(2)
75Ø PLAY"MFV15O4C2G4F8E8D8O5C4O4G2C2G2F8E8D8O5C4O4G2F2D2C2"
76Ø IF COLSEL$="Y" THEN COLOR C(4),C(2) ELSE COLOR 15,Ø
77Ø LOCATE 16,5:PRINT"PLAY AGAIN ?  Y  OR  N"
78Ø A$=INKEY$:IF A$="" THEN 78Ø
79Ø IF A$="y" OR A$="Y" THEN 1Ø
8ØØ IF A$="n" OR A$="N" THEN SYSTEM
81Ø BEEP:GOTO 77Ø
82Ø ' #### SHIELDS GONE - LOST THE GAME ####
83Ø CLS:IF COLSEL$="Y" THEN COLOR C(4),C(2) ELSE COLOR 15,Ø
84Ø LOCATE 12,5:PRINT"YOU HAVE BEEN VAPORIZED !"
85Ø PLAY"MFV15O1F2F4F16F4G+4G8G8F8F8E8F4"
86Ø LOCATE 16,5:PRINT"PLAY AGAIN ?  Y or N "
87Ø A$=INKEY$:IF A$="" THEN 87Ø
88Ø IF A$="y" OR A$="Y" THEN 1Ø
89Ø IF A$="n" OR A$="N" THEN SYSTEM
9ØØ BEEP:GOTO 86Ø
91Ø ' ############################
92Ø ' ### READ THE FILE CONFIG.WAR ###
93Ø OPEN "R",1,"CONFIG.WAR",35
94Ø FIELD 1, 3 AS A$(1), 3 AS B$(1), 3 AS A$(2), 3 AS B$(2), 3 AS A$(3), 3 AS B$
(3), 3 AS A$(4), 3 AS B$(4), 3 AS A$(5), 3 AS B$(5), 1 AS COTOG$
95Ø GET 1
96Ø FOR X = 1 TO 5:A(X)=CVI(A$(X)):B(X)=CVI(B$(X)):NEXT X
97Ø COLSEL$=COTOG$
98Ø CLOSE 1
99Ø RETURN
```

```
1000 '### NUMBER GENERATOR ###
1010 RN=A(L)
1020 LN=B(L) ' [SELECT LEVEL OF DIFFICULTY FOR RANDOM NUMBERS]
1030 N1=INT(RND(1)*RN)
1040 N2=INT(RND(1)*LN)
1050 ' ### CALCULATE CORRECT ANSWER ADDITION ###
1060 IF OP$="add" THEN CORRECT = N1+N2:RETURN
1070 ' ### CALCULATE CORRECT ANSWER SUBTRACTION ###
1080 ' ### AND SWAP TO INSURE POSITIVE ANSWER ###
1090 IF N1<N2 THEN SWAP N1, N2
1100 CORRECT=N1-N2
1110 RETURN
1120 ' ### DRAW SPACE SHIP ###
1130 DIM S(800)
1140 SCREEN 5,1
1150 LINE (5,20)-(70,20),C(1):LINE (70,20)-(70,0),C(1):LINE (70,0)-(60,10),C(1)
1160 LINE (60,10)-(15,10),C(1):LINE (15,10)-(0,18),C(1):LINE (0,18)-(5,20),C(1):
LINE (5,20)-(70,20),C(1)
1170 LINE (25,15)-(60,33),C(1):LINE (60,33)-(55,15),C(1):LINE (7,15)-(20,15),C(1
):LINE (20,15)-(20,10),C(1)
1180 PAINT (25,18),C(11),C(1):PAINT (50,25),C(11),C(1)
1190 LINE (36,20)-(56,20),C(11):PAINT(17,12),C(14),C(1)
1200 GET(0,0)-(72,40),S
1210 ' ### LASER EXPLOSION ###
1220 DIM H(800):SCREEN 5,1
1230 CIRCLE (35,20),20,C(12):PAINT (35,20),C(4),C(12)
1240 GET (0,0)-(72,40),H
1250 RETURN
1260 ' ### DRAW COCKPIT ###
1270 SCREEN 5,1:CLS
1280 LINE (0,0)-(319,199),C(9),B:LINE (5,5)-(314,194),C(9),B
1290 LINE (0,110)-(319,110),C(9):LINE (0,115)-(319,115),C(9)
1300 CIRCLE (5,110),3,C(14):CIRCLE (314,110),3,C(14)
1310 CIRCLE (48,150),18,C(9):LOCATE 23,4:PRINT"SHIELDS":LOCATE 22,27:PRINT"FIRE!
";
1320 LINE (98,135)-(128,160),C(9),B:LINE (148,135)-(178,160),C(9),B
1330 LOCATE 19,18:PRINT SG$;:LOCATE 19,24:PRINT"=";
1340 LINE (208,135)-(248,160),C(9),B:RETURN
1350 ' ### ERROR TRAP IF DATA FILE IS NOT FOUND ###
1360 BEEP:CLS
1370 SCREEN 0:WIDTH 80
1380 LOCATE 10,5:PRINT"The CONFIG.WAR data file is not on this disk !"
1390 PRINT:PRINT"USE The configuration program to create the file, then run this
 game."
1400 PRINT"THIS PROGRAM TERMINATED !":END
1410 ' ### LAST LINE OF MATH WAR PROGRAM ###
```

Listing 2:

```
1Ø '*****************************************************************************
2Ø '*                      CONFIGURE MATH WAR                                   *
3Ø '*         A COMPLIMENT PROGRAM TO GALACTIC MATH WAR                         *
4Ø '*                        ROBERT C. MILLS                                    *
5Ø '*                    3ØØ HEATHER RIDGE DRIVE                                *
6Ø '*                     FREDERICK, MD. 217Ø1                                  *
7Ø '*                       (3Ø1) 695-5228                                      *
8Ø '*****************************************************************************
9Ø ' ### INITIALIZE ###
1ØØ KEY OFF:SCREEN Ø:WIDTH 8Ø:DEFINT A-Z:FLAG=Ø
11Ø DIM A$(5),B$(5),A(5),B(5)
12Ø GOSUB 87Ø  ' [READ DATA FILE IF IT EXISTS]
13Ø ' #### INSTRUCTION PAGE ###
14Ø CLS:LOCATE 5,1Ø:PRINT" GALACTIC MATH WAR CONFIGURATION PROGRAM ":PRINT
15Ø PRINT" This portion of GALACTIC MATH WAR lets you configure the program for"
16Ø PRINT" your color or monochrome monitor and allows you to change the difficu
lty":PRINT" of the problems generated by the program in each of the five game"
17Ø PRINT" levels.  You may select values in the range of 1 - 99."
18Ø PRINT"  Please note that each game level (1-5) has an A and a B value."
19Ø PRINT" This feature allows you more flexability in designing a difficulty "
2ØØ PRINT" level."
21Ø PRINT" The A and B values represent the maximum value that the game will"
22Ø PRINT" generate for each question in the problems.          RCM85":PRINT:PR
INT:PRINT"Press  A to Abort or any other key to Begin. "
23Ø A$=INKEY$:IF A$="" THEN 23Ø
24Ø IF A$="A" OR A$="a" THEN 69Ø
25Ø '#### SHOW VALUES THAT ARE PRESET ###
26Ø CLS:LOCATE 5,5:PRINT" COLOR MONITOR  Y or N  IS NOW SET AT: ";COLSEL$
27Ø LOCATE 7,5:PRINT" GAME LEVEL         VALUE A            VALUE B"
28Ø FOR X=1 TO 5:PRINT TAB(9) X TAB(26) A(X) TAB(47) B(X):PRINT:NEXT X
29Ø PRINT"***"
3ØØ GOSUB 85Ø
31Ø IF A$="A" OR A$="a" THEN 69Ø
32Ø IF A$="C" OR A$="c" THEN 35Ø
33Ø BEEP:GOTO 29Ø
34Ø '### CHANGE MONITOR ###
35Ø CLS:LOCATE 12,5:PRINT" COLOR MONITOR  Y or N IS NOW SET AT: ";COLSEL$
36Ø FLAG=1  '  [SET FLAG SO PROGRAM WILL SAVE CHANGES]
37Ø GOSUB 85Ø
38Ø IF A$="A" OR A$="a" THEN 43Ø
39Ø IF A$="C" OR A$="c" THEN 41Ø
4ØØ BEEP:GOTO 35Ø
41Ø IF COLSEL$="Y" THEN COLSEL$="N":GOTO 35Ø
42Ø COLSEL$="Y":GOTO 35Ø
43Ø ' #### SELECT NEW VALUES FOR EACH LEVEL ###
44Ø FOR I=1 TO 5
45Ø CLS:LOCATE 12,5:PRINT" VALUES FOR GAME LEVEL ";L;" ARE SET AT:"
46Ø LOCATE 14,1Ø:PRINT" VALUE A = ";A(L)
47Ø LOCATE 16,1Ø:PRINT" VALUE B = ";B(L)
48Ø GOSUB 85Ø
49Ø IF A$="A" OR A$="a" THEN 58Ø
5ØØ IF A$="C" OR A$="c" THEN 52Ø
51Ø BEEP:GOTO 48Ø
52Ø CLS:LOCATE 1Ø,3Ø:PRINT"ENTER NEW VALUES FOR GAME LEVEL ";L
```

```
53Ø LOCATE 12,2Ø:INPUT"VALUE A = ";NVA
54Ø IF NVA<1 OR NVA>99 THEN BEEP:GOTO 52Ø
55Ø LOCATE 14,2Ø:INPUT"VALUE B = ";NVB
56Ø IF NVB<1 OR NVB>99 THEN BEEP:GOTO 55Ø
57Ø A(L)=NVA:B(L)=NVB
58Ø NEXT L
59Ø '#### SHOW VALUES THAT ARE PRESET ###
6ØØ CLS:LOCATE 5,5:PRINT" COLOR MONITOR  Y or N  IS NOW SET AT: ";COLSEL$
61Ø LOCATE 7,5:PRINT" GAME LEVEL          VALUE A            VALUE B"
62Ø FOR X=1 TO 5:PRINT TAB(9) X TAB(26) A(X) TAB(47) B(X):PRINT:NEXT X
63Ø PRINT"***"
64Ø GOSUB 85Ø
65Ø IF A$="A" OR A$="a" THEN 69Ø
66Ø IF A$="C" OR A$="c" THEN 35Ø
67Ø BEEP:GOTO 63Ø
68Ø ' ### VALUES ACCEPTED ###
69Ø IF FLAG = 1 THEN 7ØØ ELSE 74Ø
7ØØ CLS:LOCATE 12,35:PRINT"SAVING CHANGES:"
71Ø GOSUB 95Ø
72Ø CLS
73Ø '### END THIS SEGMENT ###
74Ø CLS:LOCATE 1Ø,5:PRINT"GALACTIC MATH WAR CONFIGURATION COMPLETE:"
75Ø LOCATE 12,15:PRINT" Press  R  to restart configuration."
76Ø LOCATE 14,15:PRINT" Press  G  to start the game."
77Ø LOCATE 16,15:PRINT" Press  S  to go to MS-DOS system."
78Ø LOCATE 2Ø,15:PRINT" SELECT > ";
79Ø A$=INKEY$:IF A$="" THEN 79Ø
8ØØ IF A$="R" OR A$="r" THEN 13Ø
81Ø IF A$="G" OR A$="g" THEN RUN "MATHWAR"
82Ø IF A$="S" OR A$="s" THEN SYSTEM
83Ø BEEP:GOTO 78Ø
84Ø ' ### ACCEPT OR CHANGE ###
85Ø LOCATE 2Ø,5:PRINT" PRESS  A  to ACCEPT these values or  C to CHANGE.":LOCATE
 2Ø,3
86Ø A$=INKEY$:IF A$="" THEN 86Ø ELSE RETURN
87Ø ' #### READ THE FILE CONFIG.WAR ###
88Ø OPEN "R",1,"CONFIG.WAR",35
89Ø FIELD 1, 3 AS A$(1), 3 AS B$(1), 3 AS A$(2), 3 AS B$(2), 3 AS A$(3), 3 AS B$
(3), 3 AS A$(4), 3 AS B$(4), 3 AS A$(5), 3 AS B$(5), 1 AS COTOG$
9ØØ GET 1
91Ø FOR X = 1 TO 5:A(X)=CVI(A$(X)):B(X)=CVI(B$(X)):NEXT X
92Ø COLSEL$=COTOG$
93Ø CLOSE 1
94Ø RETURN
95Ø ' #### WRITE THE FILE CONFIG.WAR ###
96Ø OPEN "R",1,"CONFIG.WAR",35
97Ø FIELD 1, 3 AS A$(1), 3 AS B$(1), 3 AS A$(2), 3 AS B$(2), 3 AS A$(3), 3 AS B$
(3), 3 AS A$(4), 3 AS B$(4), 3 AS A$(5), 3 AS B$(5), 1 AS COTOG$
98Ø FOR X=1 TO 5:LSET A$(X)=MKI$(A(X)):LSET B$(X)=MKI$(B(X)):NEXT X
99Ø LSET COTOG$=COLSEL$
1ØØØ PUT 1:CLOSE 1
1Ø1Ø RETURN
1Ø2Ø ' ### LAST LINE OF THIS PROGRAM ###
```

PCM

We edged out the competition with better solutions for every application.

Presenting the Tandy® line of PC-compatible computers from Radio Shack

At Radio Shack Computer Centers, you won't find just any computer, you'll find the *right* computer for your business. Our line of computers based on the MS™-DOS operating system lets you outfit your office with the computing power you need, without paying for capabilities you may never use. And since we carry a wide range of software and accessories, you can do all of your business at one location. So come in today and let our trained staff help you decide which Tandy computer is right for your business.

America's #1 PC compatible

Looking for a complete, low-cost MS-DOS system? Take a look at our popular Tandy 1000. Because the Tandy 1000 is compatible with the IBM® PC, there are thousands of software programs for you to choose from. But unlike IBM's PC, every Tandy 1000 comes with DeskMate® software. DeskMate includes six applications so you can start computing from day one.

The PC/XT's "mirror image"

The Tandy 1200 works like an IBM PC/XT—but costs much less. The 1200 runs the same top-name software and supports the same hardware as the PC/XT. So if you're already using one or more IBM PC's in your office, the Tandy 1200 lets you expand for less—and continue using the same software.

The graphics advantage

Searching for a quality computer with advanced graphics capabilities? Take a good look at the affordable Tandy 2000. The 2000 features high-resolution graphics to create razor-sharp graphs and diagrams in a brilliant array of colors.

Combined with computer-aided-design software, a Tandy 2000 gives your company endless opportunities for increased creativity and productivity.

The new Tandy 3000

Introducing a super-powerful computer that's the affordable alternative to the IBM PC/AT. The Tandy 3000 has all the power you need to manage your business, and it's compatible with programs designed for the PC/AT and PC/XT. The Tandy 3000 operates at 8 megahertz, twice the speed of the industry standard. It comes with 512K of main memory and ten expansion slots. And with the forthcoming XENIX™ operating system, two to six users will be able to use the Tandy 3000 simultaneously, using low-cost data terminals.

In business...for business

For the best in state-of-the-art computers, popular software, professional training, support and affordable leasing, visit your local Radio Shack Computer Center today.

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

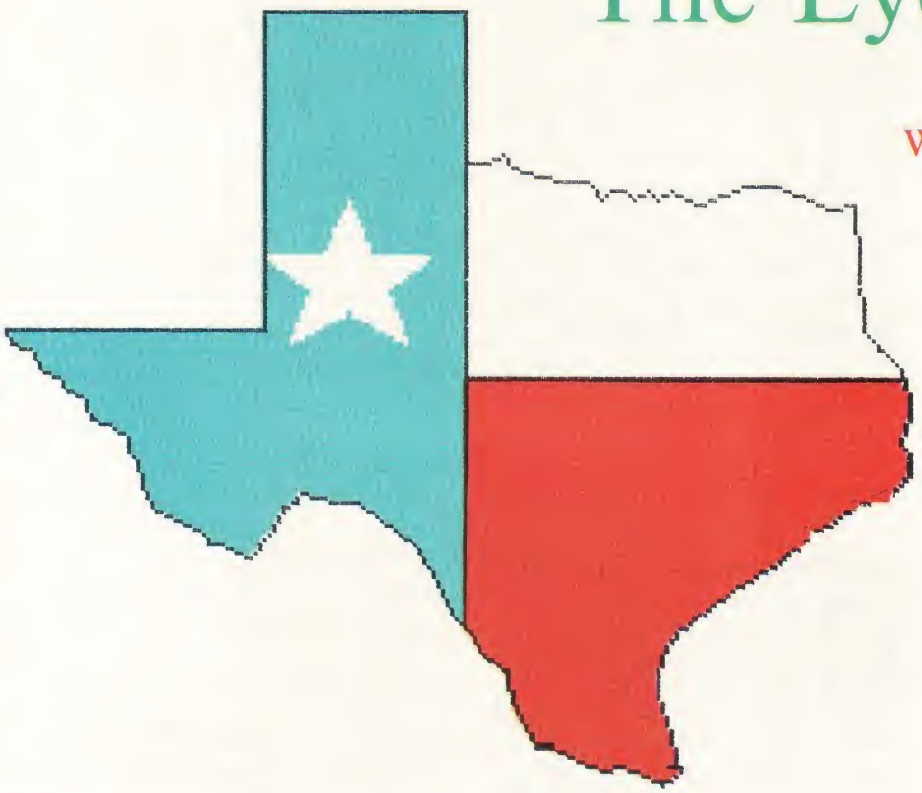
Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

New! 1986 Computer Catalog.
Mail To: Radio Shack, Dept. 86-A-138
300 One Tandy Center, Fort Worth, TX 76102

Name _____
Company _____
Address _____
City _____
State _____ ZIP _____
Phone _____

The Eyes of Texas

With Wayne Sanders, Curator



Texas, the corporate home of Tandy Corporation, mother of our beloved Tandy computers, is celebrating its Sesquicentennial this year. This month's featured Gallery exhibit runs on a Tandy 1000 and is a tribute to the state of Texas. It comes to us from Pete Martinez of Tyler, Texas.

If you would like to have your graphics creation presented here, send it in on disk with a letter describing how it works. If possible, include instructions on how to make your program work on each Tandy MS-DOS machine. A winning gallery exhibit is chosen each month and the artist is awarded $50. Address your entries to PCM Gallery, P.O. Box 385, Prospect, KY 40059.

The listing:

```
1ØØØ CLEAR ,,,32768!:CLS:KEY OFF:SCREEN 5:TS=1.745329E-Ø2
1Ø1Ø LINE (65,84)-(12Ø,84),14:FOR I=1 TO 22:READ X,Y:LINE -(X,Y),14:NEXT I
1Ø2Ø DATA 12Ø,15,163,15,163,48,167,48,171,46,173,49,183,52,189,52,192,56,194,54
1Ø3Ø DATA 197,54,199,56,2Ø2,54,2Ø4,56,2Ø7,54,213,57,223,53,229,53,239,57,241,58
1Ø4Ø DATA 246,57,248,83
1Ø5Ø CIRCLE (245,89),6,14,1Ø*TS,6Ø*TS:CIRCLE (243,1ØØ),15,14,33Ø*TS,6Ø*TS
1Ø6Ø CIRCLE (253,112),5,14,32Ø*TS,45*TS:LINE (258,115)-(254,119),14
1Ø7Ø CIRCLE (257,14Ø),22,14,1ØØ*TS,135*TS:LINE (242,125)-(238,127),14
1Ø8Ø LINE -(23Ø,134),14:LINE -(22Ø,14Ø),14:LINE -(211,147),14
1Ø9Ø CIRCLE (22Ø,16Ø),16,14,13Ø*TS,21Ø*TS:LINE (2Ø6,168)-(2Ø9,179),14
11ØØ FOR I=1 TO 41:READ X,Y:LINE -(X,Y),14:NEXT I
111Ø DATA 2Ø5,182,2Ø2,18Ø,196,178,193,179,187,177,182,174,178,174,173,164
112Ø DATA 172,159,171,156,167,154,164,149,16Ø,145,158,14Ø,152,13Ø,144,124
113Ø DATA 14Ø,121,135,121,128,119,127,121,124,122,122,127,117,133,11Ø,13Ø
114Ø DATA 1Ø8,13Ø,1Ø7,128,1ØØ,124,96,122,94,118,93,112,91,11Ø,91,1Ø7,87,1Ø4
115Ø DATA 83,1Ø3,82,1ØØ,81,98,78,97,76,94,73,93,71,9Ø,7Ø,88
116Ø CIRCLE (69,84),4,14,18Ø*TS,27Ø*TS:LINE (163,48)-(163,148),14
117Ø LINE (163,94)-(256,94),14:PAINT (16Ø,2Ø),1,14:PAINT (164,95),4,14
118Ø PAINT (164,93),15,14:LINE (141,57)-(146,69),15:LINE -(158,69),15
119Ø LINE -(149,76),15:LINE -(153,87),15:LINE -(141,8Ø),15:LINE -(129,87),15
12ØØ LINE -(133,76),15:LINE -(124,69),15:LINE -(136,69),15:LINE -(141,57),15
121Ø PAINT (141,6Ø),15:COLOR 14,Ø:LOCATE 1,1Ø:PRINT "TEXAS SESQUICENTENNIAL"
122Ø LOCATE 4,23:PRINT "1836 - 1986":LOCATE 6,4:PRINT "15Ø YEARS"
123Ø LOCATE 21,4:PRINT "THE EYES OF TEXAS":LOCATE 21,28:PRINT "ARE UPON YOU"
124Ø PLAY "MBC4F.C8F8C16F8.G16A2F2":FOR T=1 TO 6ØØØ:NEXT T:GOTO 1ØØØ
```

PCM

Making MS-DOS more useful for you

Wooing Ms.DOS

By John McCormick

MS-DOS can drive the best of us batty at times with its pipes, filters and paths, but there are some wonderful things you can do with MS-DOS if you know how.

Even more importantly, there are some things you ignore at your peril when you consign that MS-DOS manual to a dusty shelf.

I recently worked with an insurance agent who had been trying to use a Tandy 1200 for almost a full year (with little success, I might say).

I sat at his machine and typed in DIR to get an idea of his file structure before going any further. I was genuinely shocked to discover that he had *no* directories installed on his entire hard disk! The list seemed to go on forever. I asked him how he ever managed to find a file and he showed me a card where they had written the names of important files. Now that is really fighting technology, all because no one told him why he needed to use those commands in MS-DOS.

When I asked him why he hadn't, he told me he wasn't interested in any of those technical computer "things;" he just wanted to run some programs.

Just in case someone else is in that position, I will briefly explain the reason we use directories and paths.

One important reason to segregate your programs is to make your directory listing manageable. When looking for a word processing file, you don't want to sift through every *Lotus 1-2-3* or MS-DOS filename too.

Another good reason for subdirectories is to permit you to use COPY *.* to backup a set of files. This command will be a nightmare if done on a system using a hard disk with no directories, as will DEL *.*, but when used in a directory (or subdirectory) these commands will make file management much easier.

Consider this: many word processors will create automatic backup files when you save an updated version. While this can be most helpful for a short period (in case you have made a major error in the new copy), eventually your ".BAK" files will run you right out of system space.

If you are operating in a subdirectory, you can use DIR *.BAK (just to be certain), then DEL *.BAK to remove all those redundant files with ease — without any chance of deleting possibly-important system backup files (MS-DOS makes ".BAK" files of its own).

If you still aren't convinced, think about this: If you load four programs onto your hard disk (or floppy), the chances are that two to three of them will each have a file named CONFIG.SYS (and perhaps other shared file names).

"So what?," you ask. Well, MS-DOS won't permit more than one file with the same name and extension in the same directory; *that*'s what!

Just stop and think; if it *did* permit duplicate names, how would it know which one you (or a program) was referring to at a particular time?

In the case of the insurance agent, he was unable to boot MS-DOS from his hard disk because of a special program he had installed. For months he had been trying to solve this problem with no success. The local Radio Shack dealer had finally told him there was no solution, and he had been using a floppy to boot MS-DOS for six months, even though he had a perfectly good hard disk.

In 15 minutes I had cleaned the junk out of his system, built a directory tree, and installed MS-DOS in the root directory. After transferring his "problem" program to a directory, the whole system worked beautifully (and simply).

Then I left them alone for a few weeks to use their system and get used to it. I asked them to keep a log of which programs they used most often, and when I returned for another session I installed a series of batch files that would automatically change directories and start up certain programs.

The secretaries always used a word processing program, so I built a batch file for each with her name on it that loaded the word processor with just her files; likewise, the head agent only used the computer for an estimating program, and so on through the office.

Now each person gets the program he usually wants by just typing his name at the beginning of the session. This makes the computer much more user friendly (important because only one person in the office understands any of what I have done to accomplish this) and the system now is in constant use. Before I made the changes, the computer was often left off for days at a time.

I made some other changes to "sweeten" the system. Instead of C> I changed the prompt to three lines. It now looks like this:

```
Time = current time
Date = current date
C>_
```

The last major improvement I made to his system was to eliminate the possibility of making a mistake when using FORMAT (and made it easier to use in the process).

I just renamed FORMAT, then wrote a batch file to call the changed FORMAT name with Drive A: already designated.

All of these simple MS-DOS programs are illustrated following this article.

Creating directories and tree (nested)

John McCormick started programming in 1965 while majoring in physics in college, and was formerly employed with Wang Labs. He has written several reviews for THE RAINBOW.

directory structures is very simple — type MKDIR followed by your choice of name and press ENTER. MS-DOS has created the directory. For example, MKDIR WORD ENTER creates the directory WORD (perhaps a good place to put a word processor).

To create a subdirectory in WORD, just type CD \WORD ENTER and you are in WORD (an empty place to be at the moment).

To verify this, type DIR ENTER and the directory will show the path \WORD at the top. Or type CD ENTER and the screen will show \WORD.

At this point, if you do a DEL *.*, nothing will be deleted because there is nothing in \WORD.

The subdirectory is created by typing MKDIR followed by a filename and ENTER. You have created a named subdirectory inside \WORD. Typing CD\ followed by the filename (let's call it TEXT) puts you into the subdirectory TEXT.

Try typing CD ENTER. You should see \WORD\TEXT.

To put some files into this, place a disk in Drive A and type A: ENTER. (Note: you are still in \WORD\TEXT if you return to Drive C.)

Type COPY *filename* C: (where *filename* is some file on Drive A's disk).

Type C: ENTER then DIR ENTER, and you should find "*filename*" has been copied into \WORD\TEXT.

You can also stay in \WORD\TEXT and type COPY A: *filename* C: ENTER to do the same thing.

Once you are in a certain directory (or subdirectory) you stay in it when you return to that drive, until changed (CD), even if you change drives and enter a new directory on that drive. Suppose you type A: ENTER then type CD\ *dirname* ENTER (where *dirname* is a directory on the disk in Drive A). If you now type COPY *.* C: you will take all files from \ *dirname* on Disk A and copy them to \WORD\TEXT on Drive C.

Experiment a little with a new directory. As long as you don't use DEL or ERASE, there is little harm you can do. When you're done, just enter the new directory (verify with CD ENTER), then type DEL *.* to remove all the files you were playing with (although the subdirectory entries "." and ". ." will remain unless you use the RD, remove directory command).

As my teachers used to say, I will leave that to the student as an exercise. Hint: it is very simple, just look in your MS-DOS manual; it will be a good chance to see if you can still find it.

For those of you who have no confidence that you can repair any mistakes, just make a copy of both the AUTOEXEC.BAT and FORMAT.COM files on a separate disk. If you experience trouble, just delete those files in the working directory and copy the originals back onto your system disk.

Batch Files

To create a file the easy way, try this: COPY CON *your name*.BAT ENTER. This will permit you to enter file lines directly from the console (keyboard) to the file you specify in *your name*.BAT. The .BAT extension means that it will execute as soon as you type *your name* and press ENTER.

Say you have *WordStar* installed in a directory named WORD, you could type CD\ WORD ENTER then WS ENTER every time you want to use *WordStar*, or you could create a batch file to do it for you. Type:

```
COPY CON WORD.BAT ENTER
CD\WORD ENTER
WS ENTER
CTRL-Z ENTER (this closes the file).
```

MS-DOS creates a file called WORD .BAT that will change directories and start *WordStar* when you type WORD ENTER.

Suppose you had a dozen programs you want to access from the root directory (where you always start when turning on the computer). You just prepare a .BAT file for each of them giving the full path and you will save a lot of typing over the years.

This is what computers are for, to take care of the routine for you.

The AUTOEXEC.BAT is the file that makes all else simple but, before fooling with it, rename your existing file like this: REN AUTOEXEC.BAT AUTOEXEC .BAA ENTER; this will "hide" the good copy and you can rename it later (or boot from your original system disk and copy just this one file back onto the hard disk if you run into real trouble and have to reset).

Here is my AUTOEXEC.BAT and how to install it, for an example:

```
COPY CON AUTOEXEC.BAT ENTER
```

(don't type the part in parentheses)

```
ECHO OFF ENTER
```

(this stops the file from writing every move to the screen)

```
DATE ENTER
```

(this causes the date prompt at startup)

```
TIME ENTER
```

(likewise for time)

```
DIR *.BAT ENTER
```

(this shows a list of all batch files as a reminder each time the system boots — starts from scratch)

```
PROMPT Time = $T$_Date = $D$_
$N$G ENTER
```

(we'll take this one piece at a time below)

```
CTRL-Z ENTER
```

PROMPT changes the information that the computer presents each time it's ready to receive a command (normally *x*> where *x* is the current drive). I bet you didn't know you could change that, did you?

Well, it's easy. In the PROMPT line, Time = will print exactly as typed because it isn't preceded by a dollar sign. $t causes the computer to display the current time and "$_" causes the computer to go to the next screen line. Date = $d is the same as Time= $_ "$_" goes down one line, then $n causes the default drive letter to be displayed and "$g" produces the greater-than symbol (>).

My screen looks something like this when I boot my system: First I get the normal date and time prompts which I answer. Next I see a list of all ".BAT" files. Then, instead of ">" I get this prompt:

```
Time = 00:00:00
Date = DAY-00-00
C>_
```

This will continue until I type PROMPT ENTER, which will return the system to the default prompt of C>.

If you nearly always go to the same program when starting up, you can add a line for that to the AUTOEXEC.BAT file instead of entering it each time by the keyboard. You can always change your mind and go to something else; AUTOEXEC.BAT just sets up the system your way, then leaves it (and you) alone.

FORMAT is simple. Type:

```
REN FORMAT.COM AFORMAT
  .COM ENTER.
COPY CON FORMAT.BAT ENTER
AFORMAT A: ENTER
CTRL-Z ENTER
```

Now when you type: FORMAT ENTER, the batch file calls AFORMAT (the original system FORMAT command) and directs it to Drive A. You can still use FORMAT on any drive by typing: AFORMAT ENTER but, for day-to-day use when you only want to format a new floppy in Drive A, the regular FORMAT.BAT will automatically prepare to format only Drive A and will provide the usual system prompt to insert a disk in Drive A.

If you are using a two-disk system without a hard disk, just make that line AFORMAT B: to prevent you from formatting your system disk by mistake. PCM

Faster than a speeding bullet,
this program lets you fly through RAM,
leaving machine language programs in your wake.

Super Editor

By John Larrison

If you have been reading PCM lately, you may find yourself being lured into the assembly language and machine code programming of your Model 100. This short program is designed to further entice you into the depths of the machine code power of the computer.

The LCD is used very effectively to display the address, decimal, hexadecimal and ASCII character of seven consecutive bytes anywhere in the memory. You may also edit in decimal, Hex or ASCII. Other features of this program include a search routine and the ability to call a machine language routine from the display mode with one keystroke.

The Display Mode

The display mode is entered immediately following the address entry. You may enter the address in either decimal or hexadecimal. Hex numbers should be identified by an 'H' following the number. The Hex conversion routine is designed so that it is not necessary to enter lead zeros. Once in the display mode, the following options are available:

space bar	—	Display next seven bytes
hyphen key	—	Display previous seven bytes
'A'	—	Enter new address
'S'	—	Search for character string
'M'	—	Return to Model 100 menu (or use F8)
'E'	—	Edit displayed memory location
TAB	—	Execute machine code starting on first line of display

John Larrison is a service manager for the Business Systems Division of Harris/Lanier. He has developed numerous office automation and diagnostic programs currently in use nationwide.

The String Search

Press the 'S' key from the display mode and you enter this routine. You may enter any string of characters at the prompt. It may be necessary to refer to the character chart in the reference manual for the appropriate graphics characters if you need to enter non-ASCII characters. Both the character string and the starting address default to their previous values. Therefore, if you wish to find the second occurrence of a string, you need only press the 'S' key and enter twice.

The Editor

To edit the memory (got it backed up?), simply press the 'E' key. You are greeted with a unique cursor before the decimal number on the first displayed byte. If you enter a number, it is poked into this memory address. It is then read again, converted to Hex and ASCII and redisplayed. The cursor drops to the next line and reads from that address and again gives a screen display of the contents of that memory location. Although this may seem redundant, it assures that your address and actions are correct. For example, you cannot be fooled into thinking that you have edited the ROM. You are also protected from improper scrolls (remember, this is a machine code editor). For a good example of this function in operation, try poking a '1' into location F648H.

If you prefer to enter in hexadecimal notation, simply press the right arrow and the cursor moves to the Hex column. Press the right arrow again and you can edit in ASCII. It is not necessary to press the ENTER key in the

ASCII editor. You may also scroll up and down by using the cursor keys.

To return to the display mode, press the space bar from either the decimal or the Hex editor.

The Interesting Lines

12	Decimal number entry
14	Hexadecimal number entry
30-90	Display mode options
100-140	Subroutine to convert decimal to Hex and prepare display
150-180	Subroutine to convert Hex to decimal
200-260	Character string search where:
220	Sets up parameter (X) for first byte match
230	Increments address (A) and checks for end of memory
240	Detects for match of character string
260	Checks for consecutive byte match in search string (W$)
270	If complete string match is found, goes to display mode
280	If byte match is not consecutive, resets pointer (Y)
300-580	Editor where:
310-320	Cursor is generated and displayed
330	Clears cursor and checks for the ENTER key pressed
340	Looks for space bar to exit the editor
350-390	Looks for cursor keys and takes appropriate action
400	Accumulates and displays keyboard entry
410	Automatic entry if in the ASCII editor
420	ENTER key pressed from Hex editor
430	ENTER key pressed from decimal editor
440	Pokes memory, then reads and displays from same address
450	Increments pointers to next line
460	Corrects for bottom line and scroll
470	Reads and displays next line
480	Decrements pointers to previous lines
490	Corrects for top line and scroll
600-620	Subroutine to read memory, convert to Hex and display

PCM BAR CODED LISTING

The listing:

```
1 REM  ******** MCODE ********
2 REM  * MACHINE CODE EDITOR *
3 REM  *  by JOHN LARRISON   *
4 REM  ************************
10 CLS:PRINT@120,;:INPUT"Enter Address (
Dec or Hex): ";A$
12 IFINSTR(A$,"H")=0THENA=VAL(A$):GOTO16
14 A$=LEFT$(A$,LEN(A$)-1):GOSUB150:A=C
16 CLS:POKE63048,1:PRINT"ADDRESS"," DEC
  HEX  ASC":POKE63048,0
20 FOR L=0TO6:B=A+L:GOSUB600:NEXTL
30 X$=INKEY$:IFX$=""GOTO30
40 IFX$=" "THENA=A+7
50 IFX$="S"GOTO200
60 IFX$="E"THENL=0:R=0:B=A:GOTO470
70 IFX$="-"THENA=A-7
80 IFX$="M"THENMENU
85 IFASC(X$)=9THENCALLA
90 IFX$="A"GOTO10ELSEGOTO20
100 IFC<0ORC>255THENC=0:BEEP
110 X=INT(C/16):Y=C-X*16
120 X$=CHR$(X+48):IFX>9THENX$=CHR$(X+55)
130 Y$=CHR$(Y+48):IFY>9THENY$=CHR$(Y+55)
140 A$=X$+Y$:C$=STR$(C)+"   ":C$=LEFT$(C
$,4):RETURN
150 C=0:IFA$=""THENRETURN
160 FORX=LEN(A$)TO1STEP-1
170 Y=ASC(RIGHT$(A$,X))-48:IFY>9THENY=Y-
7
180 C=C+Y*16^(X-1):NEXTX:RETURN
200 CLS:Y=1:INPUT"Enter search string: "
;W$
210 INPUT"Enter start location (Dec):";A
220 X=ASC(MID$(W$,Y,1)):B=A
230 A=A+1:IFA=65535THENINPUT"NOT FOUND";
X$:GOTO10
240 IFPEEK(A)<>XGOTO230
260 IFA=B+1ORY=1THENB=A:Y=Y+1
270 IF Y=LEN(W$)+1THENA=A-LEN(W$)+1:BEEP
:GOTO16
280 IFB<>ATHENY=1
290 GOTO220
300 A$=""
310 IFX>3.9THENX=0
320 X$=INKEY$:IFX$>""GOTO330ELSEPRINT@54
+L*40+R+LEN(A$),CHR$(231+X);:X=X+.3:GOTO
310
330 PRINT@54+L*40+R," ";:IFASC(X$)=13GOT
O420
340 IFX$=" "ANDR<12GOTO30
350 IFASC(X$)=28ANDR<12THENR=R+6
360 IFASC(X$)=29ANDR>0THENR=R-6
370 IFASC(X$)=30GOTO480
380 IFASC(X$)=31GOTO450
390 IFASC(X$)<32GOTO300
400 A$=A$+X$:PRINT@54+L*40+R,A$;
410 IFR=12THENC=ASC(A$):GOTO440ELSEGOTO3
10
420 IFR=6THENGOSUB150:IFC<0ORC>255THENBE
EP:GOTO300ELSEGOTO440
430 C=VAL(A$)
440 POKEB,C:GOSUB600
450 L=L+1:B=A+L
460 IFL=7THENL=6:A=A+1:PRINT@40,;:CALL16
979
470 GOSUB600:GOTO300
480 L=L-1:B=A+L
490 IFL=-1THENL=0:A=A-1:CALL16984
500 GOTO470
600 IFB>65535THENPRINT@40+40*L,SPACE$(30
);:RETURN
610 C=PEEK(B):GOSUB100:PRINT@40+40*L,B,C
$;"   ";A$;"    ";:IFC>31THENPRINTCHR$(C
);ELSEPRINT" ";
620 RETURN
```

PCM

Using Random Files

By Alfred J. Bruey

In the August 1985 issue of PCM (Page 18), William Barden's article, "High Praise for Sequential files," described the use of sequential files. As he stated, sequential files are often the preferred way to store data. but they aren't the *only* way. Data can also be stored in random files. Before we go any further, let's look at a few definitions:

> A file is a collection of records of a particular type. A record is a collection of fields. A field is an item of information about a particular subject.

Now for an example: Let's think of a file as a file cabinet full of employee records. This file contains a record for each employee. The record contains fields: One field is the employee name; other fields might be the social security number, the hourly pay and the number of dependents.

As you can see, a group of fields about one employee comprise a record;

Alfred Bruey is a prolific author with over 30 articles published in microcomputing magazines and a book From BASIC to Fortran *published by TAB Books. He has been active in the computer field for over 20 years.*

a group of records about one company's employees comprise a file.

File Types

There are two basic types of files, random files and sequential files. To find a record on a sequential file, you must start reading the file at the beginning and read every record until you get to the one you wanted to read. Thus, if you want to read the 87th record in a sequential file, you must read the first 86 records. Furthermore, if you want to insert a record between the 86th and 87th records, you must read (and write to a new file) the first 86 records. Then the new record must be written to the end of this new file. Finally, the rest of the original file must be read in and added to the end of the new file. The old file must be deleted and the new file renamed so you'll be ready for more additions. For a few hundred records, this method isn't bad, but for a file containing thousands of records, it can become very time-consuming.

Records on a random file can be accessed in any order (randomly, as the name suggests). Thus, to access the 87th record, merely specify that you want to read the 87th record and the disk drive head goes directly to the 87th record's location and reads in the desired record.

But how do you know where a record is on a random file? You don't have to keep track of it. The system in the computer keeps track of each record's location. You can use a file creation program to generate a specific record; a month later you can ask to retrieve the same record. The BASIC program contains parameters that tell the computer where the data can be found and what format in which it is stored.

A Sample

Before we get too involved in details, let's look at an example. We'll create a random file that contains only five records, each consisting of three fields — a last name, a first name and a city name.

The program for this is shown in Listing 1. When the program is run, you are prompted to enter a last name, a first name and a city name. Be sure you have a blank, formatted disk in Drive B. The program disk should be in Drive A. Be sure you have the DOS prompt A> on the screen. Then type `BASIC B:WRITE RAN` and press ENTER. At the prompt, enter a data observation in the form of `lastname, firstname, cityname`. Be sure to press ENTER at the end of each line. Enter the following data:

```
smith, john, chicago
jones, jim, detroit
doakes, joe, pittsburgh
johnson, bill, cleveland
brown, sam, nashville
*,*,*
```

When the data is entered, type *,*,* and press ENTER. This signals the program that you are done.

Two files are created by this program:

1) RANNAMES — the file that contains the data just entered.

2) COUNT — this file contains only one value, the number of records in the file RANNAMES. Later programs will need to know this number.

Reading this Random File

The program in Listing 2 can be used to display records contained in the file RANNAMES. If you created a customer record file with a program similar to that in Listing 1, you would use a program similar to that in Listing 2 to access the data to get information about customer addresses, credit ratings, bill due dates, etc.

If you are still in the BASIC system after entering and saving this program under the name READRAN, enter

```
NEW
LOAD "B:READRAN"
RUN
```

and you will be prompted for a record number. In our particular case, a number from one to five can be entered to retrieve a record and print it on the screen, or a zero can be entered to end the look-up process.

This program has several features that are are worth noting. Note that lname$ is checked to see if it is equal to the string "***". This is the "delete" indicator for the system. Later you will see that the file edit program (Listing 3) puts a set of three asterisks in the lname$ field to indicate that the record has been deleted. Try entering various values from one to five to see how the program works.

Changing the File You Just Created

Listing 3 shows a program that can be used to update the random file, which we created earlier. To make a change to this file, or any other random file, you need only be able to perform three functions:

1) Delete a record
2) Add a record
3) Change a record

In this program, you must specify the record number of the record you wish to edit. This would not be a feasible access record for very large files. We'll talk about ways to cope with this access problem later.

Running the Program

Load the program *Editfile* into memory and type RUN. You will be prompted to enter a d, a, c or q. Be sure to use the q option to end the run because you have to let the program update the value that is stored in data set COUNT. Now let's see how the delete, add and change functions are handled by the program.

Record Deletion

You must specify the number of the record to be deleted. Note that the record is not really deleted from the file, but the value *** is placed in the lname$ field of the record. In a large system where many deletions are possible, it would be necessary to write a program that could physically remove the records from the disk and release the space for use by other records.

Adding a Record

A new record is added in the same way a record is entered in our first program — values must be entered for the three fields. The program keeps track of the record number for you. All new records are added at the end of the file, but since this is a random file these new records won't be any harder to get to than the ones that were loaded into the file earlier. The record count is updated as each new record is added and this new value is written to file COUNT at the end of the run.

Changing a Record

You must specify the number of the record to be changed. The record is displayed field-by-field on the screen. Press ENTER if you want to accept this value, or type a new value for this field before you press ENTER.

What if You Don't Know the Number?

To specify which record the user wants to delete or change, the user of the preceding program has to enter the record number. This is no problem for a file with only a few records, but for large files it can get to be a nuisance. To access a record in a large file, the user should be able to access a record by its "key." The key is the value of one of the fields (or some combination of field values) or part of one of the fields. To be more specific, let's assume for the rest of this article that the key is to be the first three characters of the last name field.

But how do we use this key? If we start looking through the random file records record-by-record using our random read routine, we would end up taking more time than if we were to stick with sequential file techniques.

What we need is a way to create a file in RAM (Random-Access Memory). This file links the keys (the first three letters of each last name in our example) with the record number in the random file. We can then look quickly through the RAM list (array) for the desired key and access the record (randomly) with the record number corresponding to that key. There is more than one way to perform this task, but I'll restrict the discussion to a simple method that is satisfactory for our system.

As you create your file using *Writeran*, you can create a sequential file,

which we will call KEYFILE, which contains only the first three characters of each last name (the first three characters of LASTNAME$). At the start of every run of program *Readran* or *Editfile*, the file KEYFILE is read into a one-dimensional array in RAM. Next the program *Readran* or *Editfile* prompts the user for the last name (the first three characters of the last name are sufficient) instead of the record number. A quick search is made through RAM to find the matching three-character key. If, for example, the three-character key MAR is found in array position 173, a random read of record number 173 brings in the desired record.

The same changes must be made in this key data as you made in the data file itself. For example, if you delete record 209, the 209th position must be changed in the RAM array to *** so the system knows later that this record has been deleted. Also, if you change the key, be sure to change the key value in RAM. (Many systems do not allow a key value to be changed; it's up to you whether you want to allow it in your system.) At the end of each run, save this key file back to disk so it is correct the next time you want to use it.

This system is simple enough to implement, but there is a complication I haven't mentioned: What if there are duplicate key values? For example, if there are 100 people with the last name Smith, you obviously have 100 key values of SMI. The search routine must be written so that if you find the value SMI, you are asked if that's the right one. If not, the program must be written so that it finds the next SMI in the RAM list and repeats the process.

This description should give enough information to get you started on using the random file system.

One Final Program

Listing 4 shows *Listfile*, a program that reads and lists the random file. Note that the record number (which is not on the file) is listed with the record. Also, deleted records are excluded from the listing. This program can be used as the basis for a fancy report generator by adding headings, tabs, etc.

A Final Note

These programs can be changed to fit your own data formats. The framework is there to handle whatever records are needed to be put on a random file. You'll probably want to add the key search routines previously described. If you do, don't forget the following points:

1) Be sure to update the key file whenever a key is updated in your data file.

2) Don't forget to write out the updated key record before leaving program *Editfile* or *Writeran* or your key file won't match the data on your random file. You might even want to add a routine to write out this key file whenever a particular key is pressed so a lot of changes aren't lost in case of a power failure. □

Listing 1:

```
10 ' readlist
20 ' read random file and list non-deleted records on printer
30 '
40 OPEN "b:rannames" AS #1 LEN=60
50 OPEN "b:count" FOR INPUT AS #2
60 INPUT#2, NUM%
70 FIELD#1, 20 AS L$ , 20 AS F$ , 20 AS C$
80 FOR I=1 TO NUM%
90   GET #1,I
100   LNAME$=L$
110   FIRST$=F$
120   CITY$=C$
130 IF LEFT$(LNAME$,3)<>"***" THEN LPRINT I;LNAME$,FIRST$,CITY$
140 NEXT I
150 PRINT "End of Run"
160 CLOSE 1
170 CLOSE 2
```

Listing 2:

```
10 ' editfile
20 ' update file: add, delete, or change records
30 '
```

```
4Ø '
5Ø '
6Ø OPEN "b:rannames" AS #1 LEN=6Ø
7Ø OPEN "b:count" FOR INPUT AS #2
8Ø ' num% is number of records in file
9Ø INPUT#2, NUM%
1ØØ FIELD#1, 2Ø AS L$ , 2Ø AS F$ , 2Ø AS C$
11Ø PRINT"(d)elete, (a)dd, or (c)hange Record or (q)uit?);
12Ø INPUT ANS$
13Ø IF ANS$="d" THEN 18Ø
14Ø IF ANS$="a" THEN 29Ø
15Ø IF ANS$="c" THEN 37Ø
16Ø IF ANS$="q" THEN 56Ø
17Ø PRINT"Press a, d, c, or q only":GOTO 11Ø
18Ø INPUT"Enter record number ";RNO%
19Ø  IF RNO%<Ø OR RNO%>NUM% THEN PRINT"Out of Range":GOTO 18Ø
2ØØ GET#1, RNO%
21Ø LNAME$="***"
22Ø FIRST$=F$
23Ø CITY$=C$
24Ø LSET L$=LNAME$
25Ø LSET F$=FIRST$
26Ø LSET C$=CITY$
27Ø PUT#1, RNO%
28Ø GOTO 11Ø
29Ø  PRINT"Enter last name, first name, and city separated by commas "
3ØØ INPUT LNAME$,FIRST$,CITY$
31Ø NUM%=NUM%+1
32Ø  LSET L$=LNAME$
33Ø  LSET F$=FIRST$
34Ø  LSET C$=CITY$
35Ø PUT#1, NUM%
36Ø GOTO 11Ø
37Ø INPUT"Enter record number ";RNO%
38Ø  IF RNO%<Ø OR RNO%>NUM% THEN PRINT"Out of Range":GOTO 37Ø
39Ø GET#1, RNO%
4ØØ LNAME$=L$
41Ø FIRST$=F$
42Ø CITY$=C$
43Ø IF LEFT$(LNAME$,3)="***" THEN PRINT"Record has been deleted ":GOTO 11Ø
44Ø PRINT "present values are "
45Ø PRINT "   Last Name is ";LNAME$
46Ø PRINT "   First Name is ";FIRST$
47Ø PRINT "   City is ";CITY$
48Ø INPUT"Enter last name ";AA$:IF AA$="" THEN AA$=LNAME$
49Ø INPUT"Enter first name ";BB$:IF BB$="" THEN BB$=FIRST$
5ØØ INPUT"Enter city ";CC$:IF CC$="" THEN CC$=CITY$
51Ø LSET L$=AA$
52Ø LSET F$=BB$
53Ø LSET C$=CC$
54Ø PUT#1, RNO%
55Ø GOTO 11Ø
56Ø PRINT "End of Run"
57Ø CLOSE 2
58Ø OPEN "b:count" FOR OUTPUT AS #2
59Ø PRINT#2,NUM%
6ØØ CLOSE 1
61Ø CLOSE 2
62Ø STOP
```

Listing 3:

```
10 ' readran
20 ' read random file
30 '
40 OPEN "b:rannames" AS #1 LEN=60
50 OPEN "b:count" FOR INPUT AS #2
60 INPUT#2, NUM%
70 FIELD#1, 20 AS L$ , 20 AS F$ , 20 AS C$
80 PRINT"Enter record number (0 to stop) ";
90 INPUT RNO%
100 IF RNO%<0 OR RNO%>NUM% THEN PRINT"Out of Range":GOTO 90
110 WHILE RNO% <> 0
120   GET #1,RNO%
130   LNAME$=L$
140   FIRST$=F$
150   CITY$=C$
160 IF LEFT$(LNAME$,3)="***" THEN PRINT"Record has been deleted":GOTO 180
170   PRINT RNO%;LNAME$,FIRST$,CITY$
180   PRINT"Enter record number (0 to stop) ";
190   INPUT RNO%
200   IF RNO%<0 OR RNO%>NUM% THEN PRINT"Out of Range":GOTO 190
210 WEND
220 PRINT "End of Run"
230 CLOSE 1
240 CLOSE 2
```

Listing 4:

```
10 ' writeran.bas
20 ' create original random file
30 '
40 '
50 ' data for file entered from keyboard
60 OPEN "b:rannames" AS #1 LEN=60
70 OPEN "b:count" FOR OUTPUT AS #2
80 NUM%=0
90 FIELD#1, 20 AS L$ , 20 AS F$ , 20 AS C$
100 PRINT"Enter last name, first name, and city separated by commas"
110 INPUT LNAME$,FIRST$,CITY$
120 WHILE LNAME$<>"*"
130   NUM%=NUM%+1
140   LSET L$=LNAME$
150   LSET F$=FIRST$
160   LSET C$=CITY$
170   PUT#1, NUM%
180   LPRINT NUM%;L$;F$;C$
190   INPUT LNAME$,FIRST$,CITY$
200 WEND
210 PRINT "File created"
220 PRINT#2,NUM%
230 CLOSE 1
240 CLOSE 2
```

PCM

Memory Upgrades for Tandy Portables!

Model 600

96K RAM Upgrade
$249. or $465. for two

Save up to **$335.** with this upgrade designed specifically for the Model 600. The 96K upgrade card fits into existing slots in the Model 600 and adds either 96K or 192K of RAM to the computer.

Model 200

24K RAM Upgrade
$89.00
Two for $170.

Save up to **$330.** with this exact replacement while bringing your Model 200 up to 72K.

Model 100, 8201 or M10

8K RAM Upgrade
$27.00
$75. for pkg. of 3

Save up to $282. upgrading an 8K Model 100 to 32K. The modules also fit the NEC 8201 and the Olivetti M10.

★ ★ ★ BUY QUALITY & SAVE ★ ★ ★

All three upgrades feature low power version of CMOS static RAM for minimum battery drain and maximum reliability. Our products are 100% tested before they leave the factory. Every upgrade carries with it a 30 day satisfaction money back guarantee and 1 yr. replacement warranty. Step by step illustrated instructions are included, and our customer service support is as close as your phone.

TO ORDER
CALL 714-540-1174
or WRITE
M/C, VISA, CHECK or MONEY ORDER
CALIF. RESIDENTS ADD 6% Sales Tax
SHIPPING-ADD
UPS Ground $1.50
2nd Day Air $4.00
Next Day Air $12.00

Tandy is a registered trademark of Tandy Corp.

American Cryptronics, Inc.

(Formerly Cryptronics, Inc.)
1580 Corporate Drive, Suite 123
Costa Mesa, California 92626
(714) 540-1174

Wrist Terminal

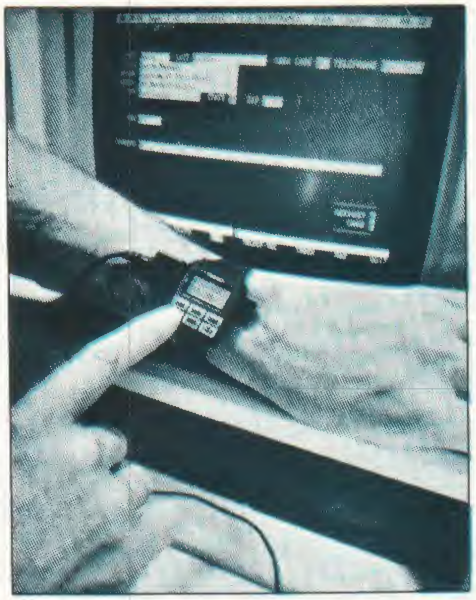
By
Carl
Oppedahl

You're at the airport with 10 minutes to spare before the flight begins boarding. Earlier that day, your watch chirped, displaying a message that it's your sister's birthday. She's in Sweden — let's see, what time is it there? Add six hours or subtract? No need to calculate, you just tap a button on your wristwatch until it shows your sister's name, and the time in Sweden is displayed. Now, what's her phone number? Tap the watch a couple more times and her phone number appears. You finish the phone conversation with a full minute to spare before boarding the plane.

Back in the office a few days later, you load a tape into the Model 100 and edit the listings for several friends' phone numbers. Hooking up a cable from the computer to the watch, you load the new information into the watch. It chirps when it is full of data. The updated information is now available on the screen.

What You Get

The watch comes with a 24-character display, two rows each of 12 characters. (The row width is just right for a phone number with area code.) Each character is formed from a five-by-seven pixel display — somewhat coarser than the six-by-eight array in the Model 100 and Tandy 200. Rather than being square, the pixels are enlongated slightly in the vertical direction, making more readable character shapes. The display shows uppercase letters, numerical digits and a few punctuation marks.



The bezel has six membrane buttons: up- and down-arrow keys that act as cursor controls, keys to select the time or alarm displays, a key for setting the time or alarm, and a key to select data display mode.

An unnecessarily large protrusion at the left side of the bezel forms the serial input contact for data loading; the RS-232 cord provided with the watch has a mating connector. The electronic "works" of the watch contain the usual lithium power cell, quartz time reference, integrated circuitry and piezoelectric beeper.

The information loaded into the watch is sent in 80 records of 24 characters, totalling just under 2K of eight-bit RAM. The watch operating system partitions RAM into a scheduled-alarm area, a world-time area and up to a dozen user-labeled data areas. The relative sizes of the areas may be changed by the user as desired; the only constraint is that at most a dozen data-area labels (each one counts as a record) may be set up and the total number of records in all areas may not exceed 80.

Software provided with the watch allows your favorite personal computer, including the Model 100 or Tandy 1000

Carl Oppedahl is a lawyer specializing in technological litigation. He is the author of Inside the TRS-80 Model 100.

to set up the data and transmit it to the watch. You type in the various pieces of information, such as:

- The date and time for one or more alarms scheduled days or weeks in advance, and a 12-character message to be displayed.
- The time zone for one or more locations, and a 12-character message to be displayed. (The city name appears as a default but you can edit it; '1' specifies the name of the person as well as the city.)
- The time and day for one or more weekly alarms, and a 12-character message to be displayed.
- 24 characters of text for each of several data entries.
- A heading (label) of up to 24 characters for each partitioned group of data entries.

When you type the information into the computer, it is kept in BASIC variables. After everything has been typed in, a few keystrokes allow you to transmit the information from the BASIC variables into the watch, destroying whatever was in the watch before.

The RS-232 link from the computer to the watch is one way; there is no way to load information from the watch back into the computer. This means you cannot, strictly speaking, edit the watch's contents. To avoid the prospect of having to retype everything whenever you want to reload the watch, the software allows you to store the data from the BASIC variables to cassette (or disk in the Tandy 1000), and vice versa. So to change one or two phone numbers in the watch, you would: 1) load the BASIC program from tape or disk; 2) reload the previously stored watch data from tape or disk; 3) edit the desired records; 4) transmit the edited information into the watch; and 5) store the edited information back onto tape or disk.

Information can reach the watch only from the provided driver program and information enters the program only from the keyboard. Don't expect to be able to load the watch based on phone numbers given in, say, your ADRS file.

Problems with the Watch

The main problem I found is with the LCD display — it is hard to get the viewing angle just right. In a market where nearly every new watch on the market, from a $30 Swatch to a $20 Casio, is water resistant, it is surprising that the RC-1000 is not. There is appar-

ently no way to silence the hourly chirping, nor to leave the daily alarm off for a day.

Once the offset for a particular time zone has been entered, the "world time" display for that zone shows a continually updated local time. This is nice, but since the watch is calculating anyway, it would have been nice if the date for the distant zone could also be calculated and displayed.

Problems with the Software

Many months ago, the Tandy 1000 was put on the market, bundled with software for the Apple, Commodore and IBM PC. I was one of the first to try out the recently released Model 100 software. It contained a number of bugs, but Seiko provides the revised program to purchasers at no charge. The Model 100 software, for example, required the user to press the ESC key at times where the documentation indicated the ENTER key was to be used. Also, some key sequences seemed to lock up the program completely, requiring a CONTROL-C, or worse, to get out. And after a CONTROL-C, all the laboriously typed watch data is lost.

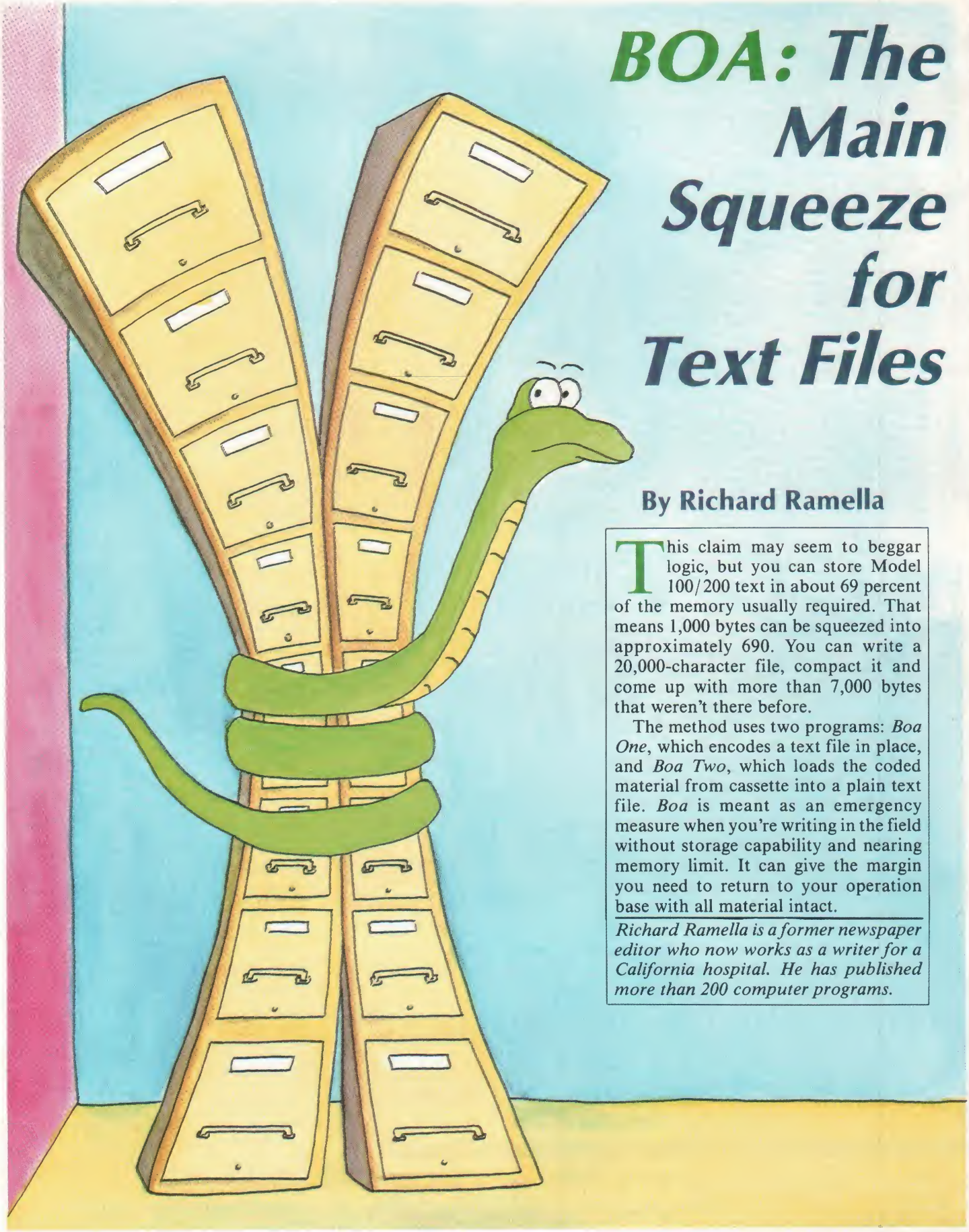
The software, written in BASIC, is so large that nearly all files in the Model 100's RAM must be deleted to fit it in. So little free space is available that the watch data cannot be sent to RAM, but rather, must go to cassette. Much of the bulk of the program comes from data lines containing various city names and time zones. I would have preferred to leave out some city name listings if it would allow the watch data to be sent to RAM.

Documentation and Support

There are user's manuals for both the watch and the software, and a handful of little slips of paper with corrections and addenda. While I found the watch manual to be clear, the software manual seemed confusing. The only way I was able to make sense of it was by loading up a sample data file included on the tape cassette and editing it. Seiko provides a toll-free number and they have a technician there who is quite knowledgeable.

Conclusion

Seiko calls this the world's smallest peripheral, and it probably is. It is a handy way to keep phone numbers and other information readily available, and surely represents the wave of the future. PCM



BOA: The Main Squeeze for Text Files

By Richard Ramella

This claim may seem to beggar logic, but you can store Model 100/200 text in about 69 percent of the memory usually required. That means 1,000 bytes can be squeezed into approximately 690. You can write a 20,000-character file, compact it and come up with more than 7,000 bytes that weren't there before.

The method uses two programs: *Boa One*, which encodes a text file in place, and *Boa Two*, which loads the coded material from cassette into a plain text file. *Boa* is meant as an emergency measure when you're writing in the field without storage capability and nearing memory limit. It can give the margin you need to return to your operation base with all material intact.

Richard Ramella is a former newspaper editor who now works as a writer for a California hospital. He has published more than 200 computer programs.

I won't keep you in suspense about how *Boa* does its work. It substitutes single, normally unused ASCII characters above CHR$(122) for high-occurrence two-character keyboard combinations. *Boa* is made efficient because it POKEs the shorter coded version of the text directly back into the file from which it's being fed. You can't see it happen, but the coded version chases the plain text, never catching up because it's truncated. What remains plain at the end is erased by you, for these are unneeded bytes. This in-place operation allows quite large files to be handled.

The Results

Since the major interest is in results, I'll cover use of the program and leave theory for last.

To test this method, create a brief text file — 80 characters or so — and run *Boa One*. At the start there's about 12 seconds of nothing, then comes a prompt: File to be BOA'd?, which must be answered with the name of the file typed in capital letters. The .DO extension need not be typed. If the file is not in the system, the program ends, but if it is found, the program has also figured out its starting address in memory. This is all that's needed. *Boa* codes the file in its peculiar way.

At the end of the run is the screen message SEE "FILE-NAME" FOR NEW VERSION. Go to the text file that is named. You will find apparent gibberish there. Press Function key 1 (F1), type *** and press ENTER. The three asterisks mark the end of the coded material. With the cursor over the first asterisk, press ENTER, press key F7 and ENTER, then press key F6 and ENTER. This erases the unneeded plain text at the end of the file. The number of characters in erased text represents the number of bytes you've just gained by squeezing the file with *Boa*.

Now you have a file filled with gibberish. The next step is to save the file onto cassette. Position a fresh cassette tape in the recorder and press the Record and Play buttons together. Go into the text file, press key F3, type the filename and press ENTER. Wait for the file to load, then rewind the tape to the start of the file and press the Play button on the recorder.

Load *Boa Two* and run it. Again there's the 12-second lull, then a prompt: Name of file to reclaim?. Answer it by typing the filename and pressing ENTER. The .DO extension need not by typed. With this the program feeds the code off the tape, turns it back into plain text and sends it to a text file in the computer. At the end of the run you will find the text file intact.

There are a few warnings. Do not type plain text onto the end of a coded file. After truncating a file, start a new one. Don't append truncated files; you could create a file that cannot be held in available memory when it's translated back into plain text. The major drawback of using *Boa* is one of time. It can take about 18 minutes to translate 16,000 characters of code from tape. I find this acceptable in the desparate times I use *Boa*. If you run *Boa One* when there are 1,200 or fewer free bytes in the system you can get an OM (out of memory) error.

The Theory

Now for some working theory.

One of the capabilities not covered in any depth by the Model 100/200 manuals is POKEing into memory. This is probably a wise omission — except to experts like you and

me — for it's possible to foul things up quite nicely when poking around in memory. *Boa One* is precise in what it does. In lines 150-190, the program searches memory for the starting address of the text file you name. Line 200 is the formula of a successful find. With this memory address stored in variable K1, the program is set to start poking the new code over the old text in lines 220-230, incrementing K1 by one with each POKE of a character into file memory.

Then there's the matter of the major magic of this system. In the English language as it is generally written, the majority of characters are spaces and lowercase letters. The four most common letters appear to be a, e, i and o, though not in that particular order. To be truthful, I inferred this knowledge from the tile distribution of a "Scrabble" set.

I decided to make a search string of fewer than 245 characters made up of most common two-character combinations occurring in general English. These are made of a space, plus letters 'a' to 'y', then each vowel mentioned, plus letters 'a' to 'y'. At the same time I created this string in the program, I made a string using most of the ASCII characters following CHR$(121). To see these strings, break into either *Boa* program when the first prompt appears, type PRINT B: PRINT E and press ENTER.

When an in-string test in *Boa One* finds a two-character string in 'B' matching what has just come out of the plain text file, it POKEs a corresponding symbol from the 'E' string back into the file as the code. If it doesn't find a match, it POKEs the left character of the search string and continues input. This is why a coded file contains capital letters, punctuation and unrecognized two-letter combinations. It also combines two spaces into one CHR$(255).

Boa Two works in reverse. If a code symbol is found in the 'E' string, its matching two characters in the 'B' string are passed on to the file. In this way the translation is made.

Further compression is possible, but I abandoned the effort when the listing for both *Boa* programs began to grow past the savings that could be realized on computers at the lower edge of memory. The longer program, *Boa Two*, requires only about 2,200 bytes to run. This in itself is part of the savings possible with the system.

Last advice: When you've used *Boa One* to squeeze out everything possible, there's one more memory grabber. Kill the listing for *Boa One* and start one last desperate text file. □

PCM BAR CODED LISTING

Listing 1:

```
100 REM * BOA ONE: Text Constrictor * TR
S-80 Model 100/200 8K
110 MAXFILES=1: CLS: CLEAR 900: DEFSTR A
-G: A=CHR$(32)+"aeio"
120 X=123: B=STRING$(2,255): E=CHR$(255)
: FOR Y=1 TO 5: FOR Z=97 TO 121
130 B=B+MID$(A,Y,1)+CHR$(Z): E=E+CHR$(X)
: X=X+1: IF X=127 OR X=143 OR X=224 THEN
 X=X+1
140 NEXT Z,Y: B=B+STRING$(2,32): E=E+CHR
$(251)
150 INPUT "File to be BOA'd";F: X=63919
160 X=X+1: IF PEEK(X)=0 THEN 160 ELSE IF
 X>64128 THEN PRINT "No such text file."
: END
170 IF PEEK(X)<>192 THEN 160
180 Q=LEN(F)-3: R=1
190 IF MID$(F,R,1)<>CHR$(PEEK(X+2+R)) TH
EN 160 ELSE IF R<Q THEN R=R+1: GOTO 190
200 K1=PEEK(X+1)+PEEK(X+2)*256
210 OPEN F FOR INPUT AS 1: CLS: PRINT "W
orking"
220 IF EOF(1) THEN 250 ELSE C=C+INPUT$(1
,1): IF LEN(C)<2 THEN 220
230 Z=INSTR(B,C): IF Z=0 OR Z/2=INT(Z/2)
 THEN 240 ELSE POKE K1+Q1,ASC(MID$(E,(Z+
1)/2,1)): Q1=Q1+1: C="": GOTO 220
240 POKE K1+Q1,ASC(LEFT$(C,1)): Q1=Q1+1:
 C=RIGHT$(C,1): GOTO 220
250 FOR K=Q1-1 TO Q1+1: POKE K1+K,42: BE
EP: NEXT: CLS: PRINT "SEE "F" FOR NEW VE
RSION.": END
```

PCM BAR CODED LISTING

Listing 2:

```
100 REM * BOA TWO: Text Reclaimer * TRS-
80 Model 100/200 8K
110 MAXFILES=2: CLS: CLEAR 900: DEFSTR A
-G: A=CHR$(32)+"aeio"
120 X=123: B=STRING$(2,255): E=CHR$(255)
: FOR Y=1 TO 5: FOR Z=97 TO 121
130 B=B+MID$(A,Y,1)+CHR$(Z): E=E+CHR$(X)
: X=X+1: IF X=127 OR X=143 OR X=224 THEN
 X=X+1
140 NEXT Z,Y: B=B+STRING$(2,32): E=E+CHR
$(251)
150 INPUT "Name of file to reclaim";F
160 OPEN "CAS:"+F+".DO" FOR INPUT AS 1:
OPEN "RAM:"+F+".DO" FOR OUTPUT AS 2
170 F=INPUT$(1,1): IF EOF(1) THEN BEEP:
END
180 Z=INSTR(E,F): IF Z>1 THEN D=MID$(B,Z
*2-1,2) ELSE D=F
190 PRINT #2,D;: GOTO 170
200 END
```

PCM

What *are* All those Strange MS-DOS Commands, Anyway?

Part II

By William Barden, Jr.
PCM Contributing Editor

In our last episode we covered some of the powerful utility programs and "batch" commands available on the Tandy 1000, 1200, 2000 and 3000. The utility programs in MS-DOS are separate program modules called by a single name from MS-DOS to perform useful functions. The batch commands actually make up a complete programming language in which a sequence of MS-DOS steps can be defined for automatic execution without operator intervention.

The SORT utility program, for example, enables you to sort any ASCII file on disk into sequence, a handy feature for alphabetizing data or directories. Last month we described how normal output to the screen or from the keyboard could be *redirected* to or from a disk file — the command `A>sort < maillist.dat >mlsorted.dat` would enter the file MAILLIST.DAT, sort it in ascending alphabetical sequence and put the output of the sorted file to the new file MLSORTED.DAT.

The SORT program can also sort in descending sequence (reverse) with its /R option:

```
A>sort /r < maillist.dat >mlsorted.dat
```

SORT can sort a file starting with any character position with its /+*n* option. This command line sorts on the fifth character position:

```
A>sort /+5 < maillist.dat >mlsorted.dat
```

Another type of redirection, called "piping," routes the output of one command to the input of another command. The following command sends the output of a directory listing to the SORT utility to produce a sorted directory and then to the MORE utility, which displays only a screen full of data before asking the poignant question - More -

```
A>dir | sort | more
```

Also described in the last column were several "batch" commands in MS-DOS — special commands that can be used to construct sequences of DOS commands that are executed automatically as batch files. Among the commands were:

- REM — Remarks

William Barden, Jr. is a master communicator in a field in which he is one of the few recognized experts — microcomputers. A prolific author of more than 27 books on computers and computer programming, Bill also has authored several instructional software projects for Tandy/Radio Shack.

- ECHO — Turns display on and off and remarks
- PAUSE — Temporarily suspends the batch operation for some manual intervention
- IF — Tests a condition
- GOTO — Alters the path of the batch commands
- FOR — Sets up repeat of commands
- SHIFT — Allows many command parameters
- SET — Changes command parameters

We showed examples of all but the last three commands. Also mentioned were "replaceable parameters" in the batch file call. Replaceable parameters are arguments that can be used within the batch file itself. The replaceable parameters take the form of variables %0, %1, %2, %3, %4, %5, %6, %7, %8 and %9 within the batch file. In the actual batch file call, they are replaced by the text used in the call.

Another utility discussed was FIND. It finds all occurrences of a given text string in a specified file. The lines of the file in which the text appears are listed on the screen. When coupled with MS-DOS batch commands, FIND can be used to create a truly impressive function. The batch file

```
CLS
echo off
echo                        ***FINDMULT***
rem   Finds file lines in which there are several fields.
rem   Syntax:  FINDMULT filespec string1 string2 string3
rem            Use dashes if no string
if exist %1 goto next1
echo File does not exist!
goto done
:next1
find /n "%2" %1 >ZZZZZ1
if %3 == - goto done
find /n "%3" ZZZZZ1 >ZZZZZ2
if %4 == - goto done
find /n "%4" ZZZZZ2 >ZZZZZ3
:done
if %4 == - goto type2
type ZZZZZ3
goto endb
:type2
if %3 == - goto type1
type ZZZZZ2
goto endb
:type1
type ZZZZZ1
:endb
echo Done with FINDMULT
echo on
```

when used with the batch file call

```
A>findmult maillist.txt Smith WI Oak
```

finds all lines containing "Smith," determines which of these lines contain "WI," then determines which of these lines contain "Oak," finally printing only those lines containing all three text strings.

Further Details about FINDMULT

Since FINDMULT is a fairly long batch file, let's look at it in more detail. First of all, FINDMULT finds up to three "subfields." It searches first for all lines containing the first text string, creating a work file called ZZZZZ1. It searches ZZZZZ1 for any second text string, creating another work file, ZZZZZ2. It searches ZZZZZ2 for any third text string, creating another work file, ZZZZZ3. The last work file is then TYPEd. From one to three text strings can be used in the search — dashes are used in place of the strings if less than three strings are specified (FINDMULT Tandy – searches only for the string "Tandy," for example).

IF commands are used within the batch file to test to see if there are more text strings. IF %3 = = - goto done for example, causes a "jump" to location DONE if the third replaceable parameter (second string) is a dash in the call. A>findmult myfile.dat Tandy - would set %0 to myfile.dat, %1 to Tandy, %2 to – and %3 to –, for example.

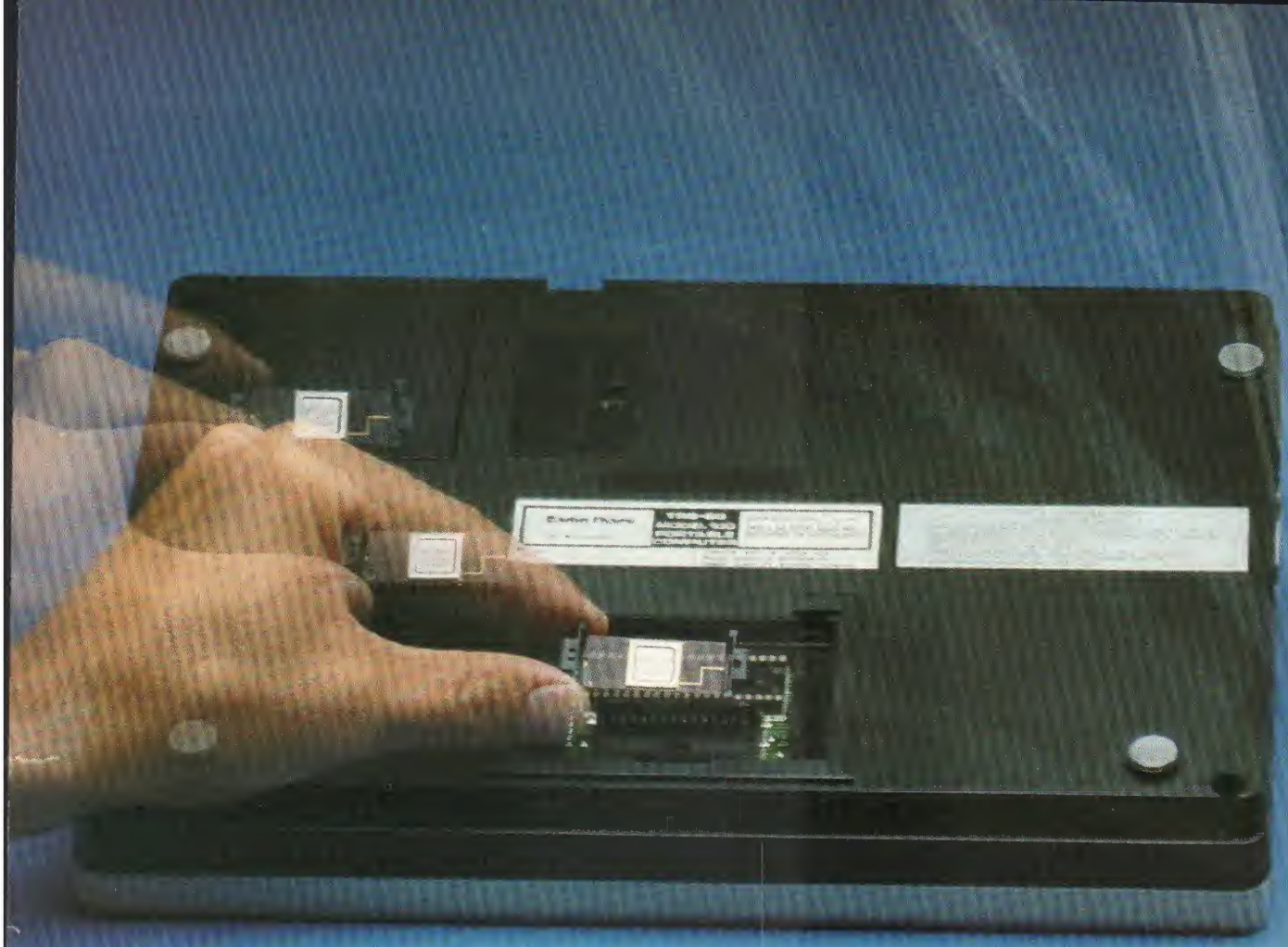
The GOTOs and labels (:next1, :done, :type2, :type1, :endb) offer a way to actually change the flow of the batch file, depending upon the program parameters.

Even though ECHO OFF disables a display of the batch file commands as they are executed, all of the text following an ECHO is displayed — a good reason for using ECHO instead of REM text, which is not displayed in the ECHO OFF case. Normal text for MS-DOS commands *is* displayed, however. A typical run of FINDMULT is shown in Listing 1.

Repetitive Operations with FOR

The batch FOR command is used to program repetitive operations in a batch file. In computerese, this type of operation is called an "iterative" operation, meaning that it has many iterations or passes. The format of FOR is FOR %f IN (set) DO . . . and a typical (non-batch) command might be A>for %f in (*.com) do dir %f.

The %f variable is like a replaceable parameter for the FOR command. The expression within parentheses defines a *set* (generally) of files for which the operation after the DO is to be done. The command above reads, "For every file in *.com, do a directory listing," and the result is the same as the separate MS-DOS commands:



IF YOU'RE MISSING THIS,

Add a ROM-Based Software Module to

If you own a Model 100, then you already know. You know that the Model 100 comes with five built-in management programs. You know about the full-size typewriter-style keyboard and the direct-connect modem and telephone auto-dialer. But did you know how easy it is to boost your portable's power with Multiplan or Interactive Solutions?

You're really missing out if you don't know how easy it is to supercharge your Model 100. Just turn your machine over, pop in either of these high-performance ROM modules, and you've got a powerful little machine that can do anything from complex spreadsheet analysis and calculations to sales forecasts, profit-and-loss projections, budgeting, pricing, engineering calculations and much more.

Powerful spreadsheet

Let Model 100 Multiplan (26-3829, $149.95) replace your pen, paper and calculator. Multiplan is an electronic worksheet—a large grid for entries, each of which can be words, titles, numbers or formulas. Multiplan then performs complex calculations . . . instantly! And because it remembers relationships between entries on a worksheet, Multiplan automatically performs calculations with a large number of variables. This lets you test plans by putting different values into your formulas.

You'll soon find that Multiplan is a vast improvement over "hand calculating." It offers a worksheet with up to 99 rows and 63 columns. You can instantly insert or erase data and widen or shrink columns, eliminating the costly and time-consuming work of typing or hand printing the worksheet over and over.

Get three programs for the price of one

Interactive Solutions for the Model 100 (26-3844, $149.95)

YOU'RE MISSING OUT!

Supercharge your Portable 100/200.

is a 3-in-1 cross-referencing software package with database management, spreadsheet and word processing.

Information stored in Data Manager, the data base, can be utilized by Data Calc, the spreadsheet program. The Word Processor allows you to merge Data Manager information into your text. Or you can select any part of a Data Calc worksheet and add it to any text file in Word Processor.

Interactive Solutions is easy to learn too, because it's designed to be self-explanatory and guide you through each process. Clear and concise labels and menus make decision making easy.

Expand your Tandy 200

Add ROM-based power to make your Tandy 200 more versatile, too. With Multi-Solutions you get three powerful applications: a word processor which allows formatting of any text file; Time Minder, a calendar program that lets you keep track of appointments; and Data Manager to help you manage dates, addresses, inventories and more.

Come in today!

Supercharge your portable today. Drop by your nearest Radio Shack Computer Center!

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Send me a new Computer Catalog.

Radio Shack, Dept. 86-A-918
300 One Tandy Center
Fort Worth, TX 76102

Name _____
Company _____
_____
Address _____
City _____
State _____ ZIP _____
Phone _____

Prices apply at Radio Shack Computer Centers and at participating stores and dealers. Multiplan/TM Microsoft Corp.

```
A>DIR SYS.COM
A>DIR TREE.COM
A>DIR RECOVER.COM
.
.
```

The asterisk character, by the way, is a "wild card" character that replaces any group of characters. The expression *.com means "all files ending in a .com extension." The expression INVENT.* means all files with the name INVENT — INVENT.BAS, INVENT.DAT, INVENT.ASM, etc.

Another wild card character is ?. It stands for "any text character in the position of the ?." The expression BILL??.DAT would include BILL01.DAT, BILL03.DAT, BILL10.DAT, and BILLAA.DAT.

The asterisk and question mark characters are used in many MS-DOS commands, and not just for batch file commands.

When FOR is used in a batch file command, then the FOR variable must have *two* '%' characters. The command above in a batch file becomes:

```
echo off
for %%f in (*.com) do dir %%f
```

The set of files enclosed by parentheses can be any number or group of files, even those specified by replaceable parameters in the batch file call:

`for %%f in (march.dat april.dat may.dat) do dir %%f`
(does directory listing for march.dat, april .dat, may.dat)

`for %%f in (%1 %2 %3) do dir %%f` (if the batch file is called by DIRF march.dat april.dat may.dat does directory listing for the three files)

The batch FOR command can be used in many useful batch operations. Take the following batch file. It's called ONLYNEW and copies files from one drive to another, but only those files that do not already exist on the destination drive. A typical call would be A>onlynew a: b:, which would copy new files from the disk in Drive a: to the disk in Drive b:

```
CLS
echo off
echo                        ***ONLYNEW***
if A%1 == A goto error
if A%2 == A goto error
echo   Copy only those source files not already on target drive.
echo   You have specified drive %1 files to be copied to drive
echo   %2. If this is correct, strike any key; if you've
echo   changed your mind, open the source drive door and press
echo   Ctrl followed by C.
pause
echo on
for %%f in (*.*) do if not exist %2%%f copy %1%%f %2%%f
goto done
:error
echo   The format is ONLYNEW A: B:, etc.
:done
```

Note that in this batch file a check is first made whether the source and destination drives have been specified. The expression `if A%1 == A` is a roundabout way of testing whether or not the replaceable parameter %1 exists. If it does not exist, A%1 is equal to A, otherwise A%1 is equal to another string, such as %a:.

The ECHO ON command is used to turn on the display so the current file specifications used are displayed. If this is not done, you'll only see the message 1 FILE(S) COPIED with no filename.

Using More than Nine Parameters with SHIFT

The batch SHIFT command permits more than nine replacement parameters (remember that the %0 replacement parameter is the name of the batch file). Its use is a little awkward, however, reminding us that the batch commands aren't a full-fledged computer language. Every time SHIFT is used, the %0 parameter is replaced by the %1 parameter, the %1 parameter is replaced by the %2 parameter and so on down the line, with the %9 parameter being replaced by nothing.

An interesting batch file that uses SHIFT is shown the listing MEMO. It is a memo file builder that takes any text and appends it to a file called MEMO.TXT. (The append operation creates a new file if none was there before, or simply adds the text to the end of the file if there was existing text.) MEMO is a kind of super simple word processor.

The batch file works like this: If MEMO alone is entered without parameters, then %1 is blank and A == A%1. In this case, if %0.txt (MEMO.TXT) exists, it is TYPEd, reproducing any previous text in MEMO.TXT. If the file does not exist, a Help message is displayed.

If there is at least one word after MEMO (not A == A%1), the commands at START are executed. First, %1 through %9 are sent to MEMO.TXT by an ECHO with redirection. Next, nine SHIFTs are done, setting %1 through %9 to the next nine words of text, or nothing, if there are no more words. Next, %1 is tested to see if it contains text; if so, the commands at LOOP are executed. This loop and output continues until the last set of nine words have been sent to the MEMO.TXT file. Because of the limitation on the maximum MS-DOS command length, only about 27 words can be sent to the MEMO file, but more can be added by additional MEMO commands.

MEMO is a handy way to jot down memos without having to load a word processor and go through an entire initialization sequence, *DeskMate* notwithstanding.

```
rem                ***MEMO***
echo OFF
if not A == A%1 goto loop
if not exist %0.txt goto none
type %0.txt
goto end
:none
echo To use MEMO, enter Memo and up to nine words of text.
goto end
:loop
echo %1 %2 %3 %4 %5 %6 %7 %8 %9 >> memo.txt
shift
shift
shift
shift
shift
shift
shift
shift
shift
if not A == A%1 goto loop
:end
```

When the text in MEMO is displayed, it is arranged as nine words per line, a fairly convenient line length. More text can be added by subsequent MEMOs.

Setting a Parameter

The last batch command is SET. SET is used to set a "global" variable (a variable available to all MS-DOS commands and programs) equal to a given character string. The variable can be used in batch files and changed with a single SET command, rather than changing many separate occurrences within all batch files. A simple example is this: Suppose you have a batch file to do a copy of all BASIC files on the diskette in Drive A: to a diskette in Drive B:. It looks like this:

```
cls
echo off
echo          ***COPY FILES***
copy a:*.BAS b:*.BAS
```

However, you would now like a batch file to copy all BASIC file copy:

```
cls
echo off
echo          ***COPY FILES***
copy a:*.ASM b:*.ASM
```

An alternative is to use a *parameter* such as ftype:

```
cls
echo off
echo          ***COPY FILES***
copy a:*.%ftype% b:*.%ftype%
```

The parameter is set off by % signs bracketing the name, which may be any character string except one containing the characters zero through nine. This parameter is really a "dummy" parameter, which is replaced by text set during a SET command, as in: A>set ftype=ASM or A>set ftype=BAS. After the SET command is executed, any batch file that refers to the named parameter has the dummy parameter replaced by the last SET command's text. This sequence:

```
A>set ftype=BAS
A>COPYF
```

will execute the batch file defined above as

```
cls
echo off
echo          ***COPY FILES***
copy a:*.BAS b:*.BAS
```

This is a trivial case of the use of SET since the batch files are so short, but in longer batch files and sequences of batch files with many occurrences of elements such as current drive specifiers and filenames to be used, SET can save a great deal of retyping and duplication of batch files. The same batch file can now work for a backup of Drive A to B or a backup of Drive C to A, depending upon the contents of a variable determined by SET.

Batch and MS-DOS Hints and Tricks

The descriptions mentioned here serve as an introduction to batch files in MS-DOS. The more you work with these creatures, though, the more little hints and tricks you will find. Here are some samples.

Sounding a Beep in Batch

Want to sound a beep at the end of a batch run, or after each separate batch operation? The BEL (bell) code in ASCII is a decimal seven, which is *not* generated from the keyboard by entering a '7', but by pressing the CTRL key, followed by the 'G' key. This sequence generates a single character file that beeps when TYPEd:

```
A>copy con: beep
<G<Z
```

The CTRL 'Z' keys terminate the keyboard input. The BEL code and other "control codes" can also be generated by certain word processors that allow embedded control codes, such as *My Word*.

Sound the beep by using this line anywhere in the batch file: `type beep`

Providing a Simple Response to a Program

When batch files invoke programs that need keyboard responses, it is possible to generate an uncomplicated sequence from a disk file by using redirection of input. The simplest case of this is just a "press any key" response for such utilities as FORMAT and DISKCOPY. As in the case of the beep file, generate a file containing a carriage return and an 'N' (for a `Format another (Y/N)` type message) by

```
A>copy con: crn (press ENTER) N<Z (press ENTER)
```

When executing a program that needs a response from the batch file, do `FORMAT A: > crn` to automatically send a carriage return and 'N' to the FORMAT (or other) prompt.

Providing a Series of Responses to a Program

The same idea can be used for more complicated responses — put the pre-canned responses in a file created from the keyboard or word processor. However, it's easy to get bogged down in complicated responses that may cause a batch file to "bomb out."

Interfacing to BASIC Programs

Batch files can be used to run complete BASIC programs if the proper files are set up beforehand for responses to questions in the program. Generally, this works best for simple programs where the responses are known beforehand. Here's a sample: The BASIC program below sets boldface printing on a Radio Shack DMP-2100 printer:

```
100 LPRINT CHR$(27);CHR$(31)
```

Not too complicated, huh? No response is required. To run this from batch, one "redirect" file is required as shown here, called RUNBAS1:

```
run "a:basic1"
system
```

This file simply runs the prdefined BASIC program and then enters a SYSTEM command to return to MS-DOS.

The batch file to run the BASIC program looks like this:

```
c:
cd\dos
basic < a:runbas1
a:
```

This batch file first sets the current drive to C:, then changes the subdirectory and loads BASIC from the subdirectory, redirecting the input to BASIC from file A:RUNBAS1. After A:RUNBAS1 returns to MS-DOS, the A: command in the batch file resets the current drive. This is a simple case, but the same basic sequence can be followed for much more elaborate BASIC files — and don't forget the system functions that can be done from within a BASIC program involving directories, subdirectories and printing.

A Date/Time Marker

If you have an MS-DOS system with a clock/calendar option, or enter the time manually, then this batch file can provide a date/time heading for files or listings. To use the file enter either `DATETIME filespec` or `DATETIME prn` / DATETIME PRN.

The batch file prints out the current date and time at the current print position of the printer if PRN or prn is specified as the file. It appends the current date and time to the end of an existing disk file. If the disk file does not exist, nothing is done.

```
rem                ***DATE/TIME***
rem    Records current date and time in file or prn.
rem    Syntax: DATETIME filespec or DATETIME PRN
date<cr >zzzzz1
time<cr >>zzzzz1
find "is" zzzzz1 > zzzzz2
if %1 == prn goto dprn
if %1 == PRN goto dprn
if not exist %1 goto done
copy %1+zzzzz2
goto done
:dprn
copy zzzzz2 prn
:done
echo on
```

This batch file uses a file called CR consisting of one ENTER character (COPY CON: CR; ENTER; F6 ENTER) to provide the carriage return for the `Enter new date/time:` prompt on the DATE and TIME command. The current date and time are first written to working file ZZZZZ1. Next, the FIND utility is used to delete the `Enter new date/time:` messages from ZZZZZ1 and to create the ZZZZZ2 file.

Depending upon the file type, printer or disk file, the ZZZZZ2 file is then sent to the printer or appended onto the end of an existing disk file by a COPY command.

Date/Time Marker on Files

The current date and time can be changed in a directory entry for a file by the simple batch file: `copy 1%+,,/B`. Use a command line such as NEWDT MYFILE.

General MORE Driver

We described the operation of the MORE MS-DOS command in the last column, but didn't provide a general form for printing any ASCII (text) file with pauses between screen pages. Here's the general form for a batch file called TMORE:

Grow up!

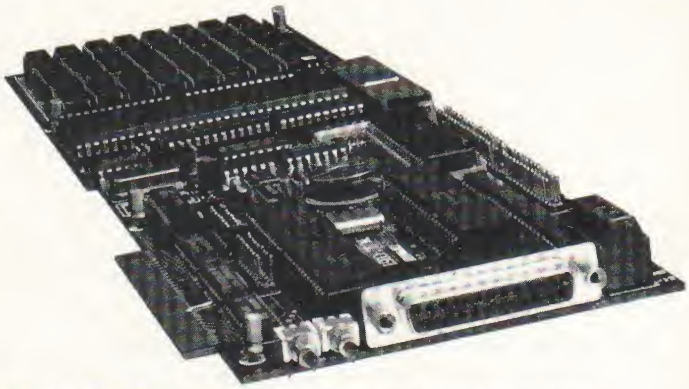
Wondering where to grow?

Why not get IBM compatibility with Tandy's 1200, or the new 3000, and take your OS-9 programs with you?

TLM's PC68K puts the 68000/68010 processor and Microware's OS-9 together in any IBM, or compatible, PC/XT/AT system, providing the best of both worlds with fully concurrent PC/68000 operation.

- Powerful languages such as BASIC09, PASCAL and C
- DYNACALC for spread sheet processing
- The power of the 68000/68010
- The price/performance and flexibility of the Tandy 1200 and 3000

TLM's PC68K . . . A great family to grow up with!



CLOCK SPEED
8/10/12.5 MHZ
Without wait states

HARDWARE
2 Parallel Ports
3-9 Serial Ports
68881 Floating point co-processor (optional)
Memory mapped dual port bus for high speed DMA transfers
Real time clock with battery backup
Up to 16 boards can run concurrently from one AT/XT/PC bus

MEMORY
1-2 MB OF RAM
(1 MB on board standard)
8K-32K of ROM
2K-8K of battery backed RAM

MULTI-USER
3-9 users (3 standard)
High speed "Tick" timer with 10ms accuracy for efficient multi-user time slicing.

OPERATING SYSTEMS
OS-9 (Multi-user UNIX look-alike)
C/PM-68K (popular single user OS)
Both support concurrent access to PC DOS

SOFTWARE
Software selectable OS environment
IBM Color/Mono Graphics Support
Local and Global disk caching for maximum speed and efficient data transfer

COMMUNICATIONS
Powerful asynchronous/synchronous data communications capability
Bisynchronous SNA 3270, X.25, etc.

MISCELLANEOUS FEATURES
Remote and Local reset (hardware or software selectable)
Remote and Local Non-maskable Interrupts (NMI)
Stand alone hardware capability (without AT/XT/PC support)

TLM Systems

West: 4704 W. Jennifer, Suite 105, Fresno, CA 93711 209/276-2345 / East: 67 Grandview Avenue, Pleasantville, NY 10570 914/747-1450

```
echo off
rem         ***TYPE WITH MORE***
rem       Types any file with ---More---
rem       Syntax: TMORE filespec
type %1 | more
echo on
```

Entering A>tmore column1.txt for example, TYPEs COLUMN1.TXT a screen at a time, then pauses with the message - - More - -, allowing you to read the screen before getting the next segment by a key press.

Changing the MS-DOS Prompt

Tired of the same dreary >A for an MS-DOS prompt? The PROMPT command in MS-DOS allows you to change the prompt to any text you'd like. Aprompt You Rang?, for example, changes the prompt to You Rang? every time the MS-DOS prompt message appears. To reset the normal default drive letter, enter prompt without any text.

```
You Rang? prompt
A>
```

If you would like trailing spaces before user command text to be entered (the space between the You Rang? and prompt in the above example), include these in the prompt message you enter.

The PROMPT command can also use a coded string variable to create a prompt message from system parameters. These codes are used:

```
$_   carriage return line feed
$b   | character
$d   current date
$e   Escape character
$g   > character
$h   backspace character
$l   < character
$n   default drive letter
$p   current directory of default drive
$q   = character
$t   current time
$v   version number
```

When used with a batch file, complicated sequences of prompts can be predefined for easy loading. Here's an example:

```
echo off
rem       Prompt Sequence 1
prompt Current date = $d$_Time = $t$_$n
```

This sequence produces a prompt similar to the following, each time MS-DOS is invoked:

```
Current date = Fri 1-17-1986
Time = 0:18:18.51
A>
```

There are plenty of other neat tricks that can be done with batch files and utility programs in MS-DOS on the Tandy 1000, 1200, 2000 and 3000. If you have a favorite batch file that you think others might be able to use, send it on to me for inclusion in future columns, either in care of PCM or P.O. Box 3568, Mission Viejo, CA 92692. PCM

Tips, Tricks and Letters

By Bobby Ballard

Since starting "The Integrated Desk" series in PCM, I've received letters from all parts of the country with questions, suggestions and advice. I want to take this opportunity to, first of all, thank you all for writing. And, I want to share some of the tips, tricks and advice I've received.

Some of the letters contained complaints or disappointments about the lack of features found in *DeskMate*. I want to emphasize again that *DeskMate* is really not intended to be an end-all program suitable for everyone. If you have very specific and powerful needs in one area or another, you might wish to consider buying a specific application program and use the balance of *DeskMate* to handle the infrequent or light-duty tasks you must accomplish.

With that said, let's get on to the rest of the mail. Guido J. Gallo wrote from Illinois to tell me about a problem in the Worksheet section on the Tandy 1000. He found it while trying to use the example in Chapter 6 of the tutorial on amortization. I duplicated his process and found the same problem. Mr. Gallo was trying to amortize a loan over 30 months; he could only get a maximum of 23 to work properly.

At first, I thought maybe the problem was due to the fact that the table is only designed to handle 12 months. However, Mr. Gallo sent in examples that proved this wasn't the problem. He had redesigned the worksheet to handle up to 30 months. I then designed a worksheet that should have amortized an amount over 60 months or five years. I ran into the same problem at that point. Anytime I entered an amount over 23 for the number of periods, I got an OVERROR printed in a massive number of cells. This problem seems to be well-documented at Tandy since several wrote to me describing it and indicating that they had informed Tandy as well.

I contacted Tandy to see if they had a solution for this problem and was informed that they don't consider this to be a bug, but a limitation of *DeskMate*'s design. So the ability to enter an amount over 23, for amortizing, is not possible and is not a planned fix. If anyone knows of a way to get around this, I would like to hear from you so the solution can be shared with everyone. If anyone passes on a solution I will let everyone know immediately on Delphi, and as soon as possible here in the column.

One other source of confusion pointed out by a couple of readers concerns the use of the SHIFT PRINT function in the Tandy 1000 version. In *DeskMate* you have the option of printing a screen dump or a file. The SHIFT PRINT function is used to do a screen dump. When invoked, it prints everything on the video screen including the menus, time and date, and prompts; everything. If you want to quickly get a printout of only one record in *Filer* or just a small section of text in the *Text* section, then SHIFT PRINT is a good choice to use.

However, if you want to print a file or entire document, you must use the print function. The print function can be invoked in two ways: on the menu, when it's available, at the bottom of the screen, or by pressing the PRINT key on some models or the PRT SCR key on other models. Also, make sure you have previously set the print parameters correctly by using the ALT F6 function from the main menu.

I also heard from someone who was having trouble printing on legal paper (8.5 inches wide by 14 inches long). I tried this and had no problem. You must remember to set the printer parameters using the ALT F6 key. I successfully accomplished legal-size printing by setting the number of lines per page to 84 and the number of printed lines per page to 78. The first number is arrived at by adding 66, the number of lines on letter size paper to the added number of lines for legal, which is 18. There are normally six lines to an inch, so I multiplied six by the three additional inches for a total of 18. That gives a total of 84. I then took off six for two margins of three lines each. This left 78, the second number, for printed lines per page.

This same letter indicated the user was having problems with blank lines that weren't on the screen periodically showing up on a printout. To avoid this, you must make sure the screen format, the F4 key in *Text*, is set the same as the printer parameters using ALT F6. If you have lines longer than the printer margins are set, you will also get a wrap-around of text that causes blank lines or half-full lines of text to appear in the printout. Make sure your printer and printer settings in *DeskMate* and the actual file all have the same format. This should take care of blank lines and more.

Speaking of printers, keep this tip in mind. If you wish to have your printer configured a certain way each time *DeskMate* is booted, make sure you exit *DeskMate* with the settings you want already selected. Then use the F12 key

Bobby Ballard is a free-lance writer and the owner of a computer software and consulting firm. He also operates a BBS in Brooklyn. Bobby can be contacted at 1207 Eighth Avenue, Apt. 4R, Brooklyn, NY 11215.


TANDY 600 PORTABLE COMPUTER
ESC
TAB
CTRL
SHIFT
ENTER
ALT
NUM
LABEL

Meet the disk-based Tandy 600.

A breakthrough in portable computing power and versatility.

Get a portable that matches the performance of a desktop computer. The amazing Tandy 600 features a 16-bit microprocessor, an 80-character by 16-line display, a built-in 3½" disk drive that stores 360K of data and 32K RAM (expandable to 224K*).

Five resident applications

With the Tandy 600's larger display and expanded memory, Multiplan spreadsheet analysis can hold more information. Word processing is as easy as using MS-Word. You'll have quicker access to documents, and better storage with the built-in disk drive. File is an electronic data base for names and addresses, expenses, client billing, inventory and more. And you can keep a large number of different files on the pocket-size diskettes. With Telecom and the Tandy 600's built-in modem, you're able to communicate with other computers over phone lines and access national information networks. Telecom will even dial the phone number of anyone listed in the File program. Calendar helps you keep track of your daily tasks and activities.

The easy-to-learn resident System Manager lets you run each application, exchange information between applications and manage the files created. And you can add BASIC/ROM (26-3904, $129.95) to write your own programs.

Tandy . . . Clearly Superior™

If you've been looking at portable computers, compare them with the Tandy 600 (26-3901, $1599). You'll be amazed at what it can do!

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Send me a free Tandy 600 brochure.

Radio Shack, Dept. 86-A-1127
300 One Tandy Center
Fort Worth, TX 76102

Name _____
Company _____
_____
Address _____
City _____
State _____
Zip _____
Phone _____

* RAM upgrades are in banks of 96K (26-3910, $399.95 each). Prices apply at Radio Shack Computer Centers and participating stores and dealers. Multiplan and MS-Word/TM Microsoft Corp.

to exit back to DOS. If you don't wish to save the current settings for the printer, use the SHIFT F12 to exit. This preserves your version of *DeskMate* as it first booted. This is true of the screen colors, too.

A Tip for a Mouse

If you have the Tandy Digital Mouse installed on your system and find that the *Telecom* function will not work properly, then you have to change the way you boot *DeskMate*. Apparently the mouse driver software affects *DeskMate*, particularly in the *Telecom* section. Tandy is aware of this problem and is working to fix the bugs. The fix will probably involve changes in the MOUSE.SYS software instead of *DeskMate*, according to a spokesperson at Tandy.

To avoid this problem involves some choices on your part, at least until the fix is released. You can boot *DeskMate* without the mouse driver as part of the CONFIG.SYS file. This, of course, means you won't have access to the hardware clock on the mouse controller board. This is due to the fact that the mouse driver (MOUSE.SYS) must be installed and executed in order to install and execute the clock software.

Another solution to this problem is to get a second clock on your next board and use the software for that clock at all times. You can then set up a bootable copy of *DeskMate* that does not include the mouse driver software. This is the way my system is configured and it works well. I use the mouse board just for the mouse and use the clock on a multifunction board as my timepiece.

Anyway, it's up to you as to what route to go in solving this problem. The least expensive solution is to boot without the MOUSE.SYS software installed. But if you're adding another board soon, a clock is available on many and adds little to the overall costs.

While we're covering communications and hardware, let me remind you that *DeskMate* only works on COM Port 1 of your MS-DOS computer. When adding COM Ports, make sure you leave your modem configured or attached to COM Port 1 or else *DeskMate* won't run at all in *Telecom*.

***Worksheet* Help**

Richard Wagner wrote a long letter from Oregon with several observations and a question on using the spreadsheet section. He writes, "I would like to take a column of my prices and add or subtract a specific amount from every item in the column, say 50 cents or $1.25. Can anything be done?"

To do this, you might set up a column of formulas, using the formula and select functions, that adds the amount from an input prompt. Use the formula function to define a cell as ?INPUT, then add the amount to an entire column. This should allow you to add a specific amount at anytime.

Again, if anyone knows of a better way to solve this problem for Mr. Wagner, please let all of us know. In the near future, I will cover the *Worksheet* function in more detail.

Mr. Wagner also writes, "Can't figure out how to underline . . ." This becomes a problem without the ability to embed printer codes in *Text* or other functions. Underlining takes place by sending special escape codes to the printer to turn underlining on and off. *DeskMate*, at least on the Tandy 1000, does not support embedded printer codes, so underlining is difficult if not impractical. You could turn the underline on from the DOS command level before entering *DeskMate*, but then everything will be underlined.

Several readers have asked about the slow movement in the *Worksheet* section while going from one column to another or scrolling to the end of a large spreadsheet. This may not be a problem on the versions that run on faster machines like the Tandy 2000 or Tandy 3000. The 1000 has a slower clock speed so the program runs slower. Don't forget, though, that using SHIFT or CTRL along with the arrow keys jumps you around in the spreadsheet much faster. To move a screen at a time, use SHIFT and the arrow keys and to jump from the first column/row to the last column/row using the CTRL and arrow keys.

While we're on the subject of *Worksheet*, I have one other small solution to a problem with using large spreadsheets. Several have asked about not being able to know or see the row or column name while out in the middle of the sheet. One guy said he just wrote the names down on paper before moving about. If you have a printer, you could use the SHIFT PRINT function discussed earlier to get a quick printout of column or row names. Another solution that might work is to embed names of the column and rows somewhere in the spreadsheet. You could make additional title cells for working in the spreadsheet and delete them before printing. It would leave a blank row or column in the sheet, but this might not matter on some occasions when just getting the job done is more important.

I also have a report on two bugs from Tandy Towers and they assure me they are working to fix the following two problems. (These are to be fixed in the next version and have been confirmed by Tandy.) Both of these bugs are in the *Worksheet* section of *DeskMate*. The first one concerns setting up a formula that raises a negative number to a given power. Currently this process returns a one (1) instead of the correct answer. I haven't been able to think of a way around this until the new version arrives.

The second bug concerns the way in which F10 inserts in the *Worksheet*. The F10 key presently moves the data up, above the insert, and replaces the figures in the cells. This will also be fixed in the next version.

Merging Mail

I've had several people ask me how to get *DeskMate* (on a 1000) to do a mail merge between *Filer* and *Text*. I had to inform them that I knew of no way to accomplish this task. The problem stems from the inability of *DeskMate* to handle variable fields in the *Text* section, and the additional programming it would take to pull information from the *Filer* section and move it to *Text* automatically.

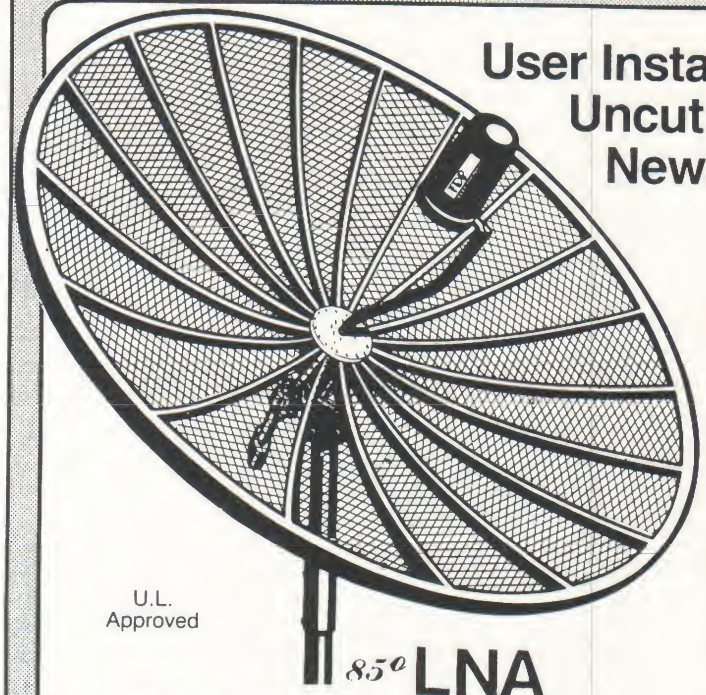
It is easy to move a single block of data and the tutorial gives a pretty complete explanation. It must be done manually, however, and if you need to handle a large mailing it is impractical. It would be nice to know someone is working to make *DeskMate* do a mail merge, but I've not heard anything so far.

The Wrap Up

I hope this information has been helpful or, even better, helped you solve a problem you have been experiencing. If you have any other questions, problems or fixes, just write me at 1207 Eighth Avenue, Brooklyn, NY 11215 or for quicker help and response, contact me on Delphi in the MS-DOS special interest group. Address mail on Delphi to my user name, BOBBYBALLARD. PCM

Discover a Whole New World of TV Entertainment!

SATELLITE SYSTEM



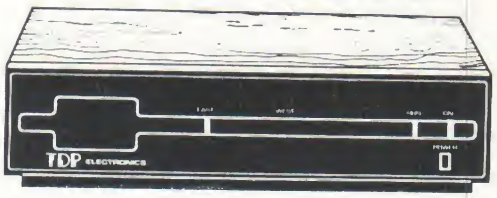
U.L. Approved

85° LNA

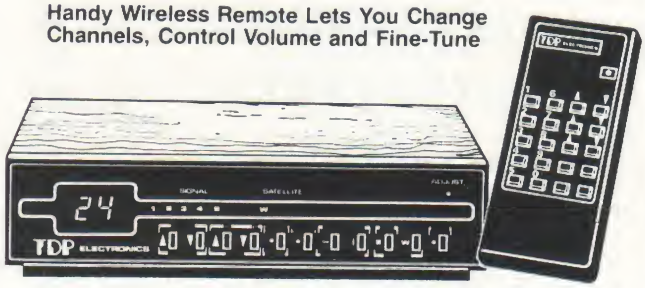
User Installable! Over 100 Channels! Uncut Movies! Specials! Sports! News! Weather! Music Videos!

Entire System **$1519**

- Totally Automatic System With 8½ -Ft. Dish, Receiver and Wireless Remote Control
- Comes With All Accessories Needed for Complete Do-it-Yourself Installation
- Descrambler Ready



Handy Wireless Remote Lets You Change Channels, Control Volume and Fine-Tune



Now you can enjoy it all! 24-hour non-stop entertainment that you simply can't get from local TV, cable, and even video tape! You'll see the hottest stars, recent uncut and uninterrupted movies, the ultimate in sports coverage—and the best part is, there are no monthly fees to pay. You'll see exciting family-oriented programs, plus music videos, FM broadcasts and movies with provision for stereo, and 24-hour news coverage on both TV and radio. It all adds up to out-of-this-world TV viewing at a down-to-earth price. Come in today and discover how easy it is to enjoy satellite programming! Includes complete installation manual.

(817) 825-4027

NOCONA ELECTRONICS • Box 593 • Nocona, TX 76255

UPS Shipping

The Delphi MS-DOS SIG has really been picking up steam over the past month. Membership is now well over 2,000, with as many as 100 members accessing the SIG on a single day. And even better, more and more knowledgeable Tandy users are taking the time to participate, share their programs and help others with answers and advice.

The databases are beginning to fill with public domain and user-supported, or "freeware," programs. There are many excellent utilities, full-featured terminal programs, games, home and business programs and programming aids. If you are not familiar with user-supported software, these are market-quality programs, distributed freely by the producers with a note asking for a small donation if you like and use the software. It is a great marketing concept, giving you the chance to use a program before paying for it and also saving you the tremendous distribution costs tacked onto conventionally distributed software products.

Regretfully, Delphi has instituted an hourly rate hike, effective March 3. The basic rate for evening and weekend hours, previously $6, will increase to $7.20 an hour for mainland U.S. and Canadian members who access via Uninet and Tymnet. The prime daytime rate will increase from $16 an hour to $17.40.

Other rates apply for Hawaii, Alaska, Puerto Rico and international users. More information about these may be obtained online in the MS-DOS SIG or by calling Delphi at 1-800-544-4005.

There is some good news regarding rates on the horizon, however. Beginning March 3, Delphi will levy no extra charge for 2400 Baud access. Other online services charge premium fees for access at higher speeds, some as much as two or three times their standard rates. Also, they are beginning a value-packed membership plan called "The Delphi Advantage" which offers lower access rates, free manuals and command cards, a newsletter and other benefits. In exchange, members agree to use, essentially, six hours of online time each month at a guaranteed $6 an hour basic rate.

Other good news involves enhancements in the Delphi software. Commands and features are constantly being added to make the system more flexible, powerful and user-friendly. Support for the Kermit file transfer protocol has recently been added to the workspace area and should soon be available in the open databases. This offers several advantages over Xmodem, including greater reliability of data transfer, multiple file transfers, and seven-bit transfers rather than eight (which should be a boon to our Canadian members who access through the seven-bit Datapac network).

At any rate, if you haven't joined us in the MS-DOS SIG by this time, I hope you will soon. Telecommunications is one of the most interesting and exciting things you can do with your computer. At the risk of sounding trite, I have to say it: There really *is* a whole world out there waiting.

By Kevin Nickols
MS-DOS SIG Manager

Submitting Material To PCM

Contributions to PCM are welcome from everyone. We like to run a variety of programs that will be useful/helpful/fun for other Tandy Portable and MS-DOS computer owners. We now support the Tandy portable models 100, 200 and 600 and the Tandy 1000, 1200, 2000 and 3000 MS-DOS computers.

Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs. All programs should be supported by some editorial commentary explaining how the program works. Generally, we're much more interested in how your submission works and runs than how you developed it. Programs should be learning experiences.

Pay for submissions is based on a number of criteria. The rate of remuneration will be established and agreed upon prior to publication.

For the benefit of those who wish more detailed information on making submissions, please send an SASE to: Submissions Editor, PCM, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit programs or articles currently submitted to another publication.

If you feel qualified to review software and/or hardware products for computers covered in PCM, send us your name, address and phone number; we will send you a questionnaire form and a copy of our reviewer guidelines.

DELPHI™

This abbreviated, modified version of Delphi's command card has been created to help our readers who use Tandy® MS-DOS Computers get started quickly on PCM's new MSDOS SIG. It is being reproduced here for your convenience and can be removed, if you wish, and kept near your computer for easy reference.

WELCOME TO DELPHI

Most Delphi commands are self-explanatory. This card will serve as a handy backup reference.

Signing onto Delphi Directly

1. Dial (617)-576-0862.
2. When you have carrier, press [ENTER] once or twice.
3. At "USERNAME" type your membername and [ENTER].
4. At "PASSWORD" type your password and [ENTER].

How To Sign On Using Uninet

1. Dial your local Uninet number.
2. Hit [ENTER] [.] [ENTER] at the |x| or "L?" prompt.
3. Type DELPHI or GVC at the SERVICE prompt.
4. Then type your USERNAME and PASSWORD as outlined above.

How To Sign On Using Tymnet

1. Dial your local Tymnet number.
2. When "PLEASE TYPE YOUR TERMINAL IDENTIFIER" appears, type A.
3. When "PLEASE LOG IN" appears, type DELPHI.
4. Then type your USERNAME and PASSWORD as outlined above.

How To Sign On Using Datapac (Canada)

1. Dial your local Datapac number.
2. Type [.] for 300 baud or [.] [.] for 1200 baud.
3. Type Set 2:1, 3:126 for full duplex allowing deletes.
4. Type p 1 3106, DELPHI; [ENTER] (Tymnet)
5. Then type your USERNAME and PASSWORD as outlined above.

To obtain your local access number you may call Tymnet at 800-336-0149 or Uninet at 800-821-5340. If you have problems at any time, call Delphi toll-free at 1-800-544-4005. (Mass. 617-491-3393)

Note: Most commands require only enough letters to be entered to make them unique. For example, to enter CONFERENCE from the Main Menu, simply type "C" and [ENTER]. Do not press [ENTER] after commands using the Control Keys. Most other commands require pressing [ENTER] to activate them.

Typing BYE from any prompt (except the MAIL prompt) will log you off of Delphi.

Typing [?] [ENTER] will generally display a full menu or provide help.

IMMEDIATE COMMANDS (Can be used at any time.)

/HELP — lists Immediate Commands.
/ECHO — turn on character echo.
/NOECHO — turn off character echo. (Used after setting terminal or Uninet or Tymnet node to produce echo.)
/EXIT — exit to next higher menu or command level.
/GAG — turn off incoming /PAGE or /SEND messages.
/LENGTH — shows current number of lines per page on your screen or sets new length.
/NOGAG — turn on incoming /PAGE or /SEND messages after using /GAG.
/PROMPT (1, 2, or 3) — 1=no menu, no explanation; 2=no menu, some explanation; 3=menu plus explanation.
/TIME — show current Eastern time and date.
/WHOIS (username) — shows profile of member (if available).
/WIDTH — shows current screen width format or sets new width.

DATABASE

DIRECTORY — display a directory of all files in the topic.
EXIT — exit database.
HELP — get help on database actions and commands.
READ — read a description of a file. (You must read the file before you download it.)
SEARCH — search a topic by keyword.
SET TOPIC — switch from one topic to another without leaving the database section.
SUBMIT — submit a file for inclusion in a topic. The file must be in your workspace.
WORKSPACE — enter your workspace area.

In order to access a file, you must first READ (filename). Once you have read a file, the following actions are available:

DESCRIPTION — displays the file's description again.
DISPLAY — display/list the file on your screen.
DOWNLOAD — use with the buffer capture method of downloading.
EXIT — return to the database prompt.
HELP — get help on commands and actions.
LIST — like display; list a file in an unformatted format.
NEXT — advance to the next group or file. ([ENTER] defaults to NEXT.)
XMODEM DOWNLOAD — download the file using the Xmodem protocol.

WORKSPACE

Workspace is an area for you to store files and messages of all types. This is where you must first upload a file before submitting it to a database. You can file forum messages for retrieval later. Mail messages can be stored here.

From the MSDOS SIG> prompt type DA, and pick a topic, then type WO to reach Workspace.

APPEND — append one file to another.
CATALOG — shows which files you have created.
COMMON — go to the Delphi Common work area.
CREATE — creates file and stores it in your area.
DELETE — deletes files you no longer need.
DOWNLOAD — download a file from Delphi to your disk.
EDIT — create and edit your own text files.
EXIT — return to Main Menu.
HELP — explanation of WORKSPACE commands.
HOME — return home to your private work area.
LIST — lists contents of any file in your catalog.
PUBLISH — submits your file for Delphi publication in Authors, Members Choice or Newsletters.
PURGE — delete all but current version of duplicate files.
UPLOAD — upload a file from your computer to Delphi.
XDOWNLOAD — download via XMODEM protocol.
XUPLOAD — upload via XMODEM protocol.

MSDOS SIG FORUM

ADD — start a new message thread with a different topic.
BACK — moves backwards within a thread.
DELETE — delete a message.
DIRECTORY — display a directory of messages.
EDIT — edit the current message.
EXIT — exit forum.
FILE — put a copy of a message in your workspace.
FOLLOW — follow a message thread. Read only the messages of a particular thread.
FORWARD — send a copy of a message by mail.
HELP — get help on forum actions and commands.
HIGH — set/show the high message number.
MAIL — take you directly to mail.
NEXT — read next message. ([ENTER] defaults to NEXT.)
READ — read a message. (Typing message number will read that message.)
REPLY — reply to a message.
TOPICS — set/show message topic.

CONFERENCE

EXIT — return to Main Menu.
JOIN (groupname) — join existing group or start new one.
NAME (newname) — change your name or "handle".
PAGE (username) — pages another user in the system.
SCHEDULE — transfer you to the Conference Schedule.
WHO — lists all current users and Conference groups.
Conference Immediate Commands (use while in Conf).
/ACCEPT — accept another's page from within current group.
/ANSWER — respond to or decline PAGE from another user.
/CANCEL — terminate a PAGE to another user.
/EXIT — like CONTROL-Z; gets you out of wherever you are.
/GAG — disable /SEND's from people outside your conference group.*

/GLOCK — lock the group's attributes.*
/GNAME (newname) — change current group name.
/GPASS (password) — select a group password.*
/GPRIVATE — make the group private.*
/GQUIET — makes the group have silent entry and exit.*
/HELP — get help on conference actions and commands.*
/JOIN (groupname) — join an existing group.
/LOG — save a transcript of your conference in your workspace.*
/MAIL — takes you directly to Mail.
/NAME — create a conference nickname (handle).
/PAGE — ask another user to join your group.
/PASS (password) — say the password for admittance into password groups.
/REJECT — a pleasant "No thank you" to whomever is paging.
/REPEAT — turns Echo on or off.*
/RNAME (nickname) — show the username of a person using a handle.
/SCHEDULE — transfer you to the Conference Schedule.
/SEND (username) — send message to current user.
/SQUELCH (username) — ignore messages from a user.*
/TALK — like /JOIN, but doesn't leave current group. /# also works, where # is the number of a conference group. Allows you to participate in more than one group at the same time.
/WHO — lists all current users and Conference groups.
/WHOIS (username) — displays (username) Profile.

*Note: Many of these commands may be preceded by NO. For example, /GAG disables sends, but /NOGAG resumes them.

DELPHI MAIL

Primary Mail Menu (DMAIL)
CATALOG — lists all Mail files you have created.
EXIT — return to Main Menu.
HELP — explanation of Mail commands.
MAIL — send or read mail. Enters Secondary Mail Menu.
SCAN — display the headers for all unread mail.

Secondary Mail Menu (MAIL)
[ENTER] — depressing the return or carriage return key, will read the next message or more of the current message.
BACK — displays previous message.
DELETE — deletes current (last read) message.
DIRECTORY — lists summary of your mail messages.
DIRECTORY / Folder — lists folder names.
DIRECTORY (folder name) — lists summary of messages in the specified folder. For instance, DIR Pending.
EXIT — returns to Main Menu.
EXTRACT (filename) — adds current message to named file.
FILE (folder name) — adds current message to the named folder.
FORWARD — forward present message to others.
NEXT — skips to next Mail message. ([ENTER] defaults to NEXT.)
READ — displays your Mail messages.
READ (folder name) — reads contents of named Mail folder.
READ (n) — allows you to read selected message number.
READ /NEW — for new MAIL arriving while in MAIL.

REPLY — sends a reply to sender of current message.
SEARCH (string) — searches current Mail file for specified character string.
SELECT — pick messages for delete operation.
SEND — sends message to another user or users.
SEND (filename) — sends file (filename) to other user(s).
SEND/EDIT — calls editor to edit message being sent.
SEND/LAST — uses last message as text for current message.

HELP

Contains a full description of all Delphi services using the same structure as the Delphi Menus.

PEOPLE ON DELPHI

Enter information about yourself; find out about others.
I-AM — add or change information about yourself.
ADD — adds to existing information.
CHANGE — removes all current information about you and request new info.
DELETE — deletes all information under a given keyword heading.
DISPLAY — prints your personal profile.
EXIT — returns to Main Menu.
WHOIS (membername) — displays member profile if available.
SEARCH — find members with particular interests.
BROWSE — browse through member profiles.
LIST-KEYWORDS — shows keywords used in member profiles.

This section is accessed from the Delphi Main Menu. When someone does a /W command in conference, this is the information that will be displayed about a member.

USING-DELPHI

ADVICE FROM DELPHI — answers to most frequently asked questions.
CREDIT POLICY — explanation of current DELPHI policy.
GUIDED-TOUR — a brief version of the tour you took at signon.
MAIL TO SERVICE — send comments and suggestions to DELPHI.
NETWORK-INFO — phone numbers and login procedures for data networks.
PREMIUM-SERVICES — information concerning the extra cost services.
RATES-AND-PRICES — official Delphi rates and prices.
SETUP — terminal and network configuration.
LENGTH — lets you find your screen length and tailor Delphi accordingly.
MENU — choose default menu at sign-on.
PASSWORD — change your password. (frequent changes are recommended.)
PROMPT — select level of menu prompting desired.
SET-TYMNET — experiment with setting network parameters.
TERMINAL — special features for DEC VT100 and VT52 users.
WIDTH — tailor Delphi to fit your screen width.
USAGE-HISTORY — view your to-date activities on Delphi.

DELPHI TERMINAL CONFIGURATION GUIDE

8 bit ASCII*
1 stop bit*
no parity*
asynchronous
full-duplex
no auto-linefeed or carriage-return linefeed
XON-XOFF or Handshaking should be enabled

*sometimes you have to experiment with other combinations such as: (7 bit, 1 stop, noparity) or (8 bit, 1 stop, even or odd parity).

NOTES

To erase a character, Delphi uses the ASCII delete/rubout key which is decimal 127. If necessary, the terminal program should translate the backspace key to a delete/rubout. Unfortunately the networks do not echo the delete/rubout correctly however it will have the desired effect.

Delphi uses the following control characters:

CONTROL-Z — end of input or exit to next higher menu.
CONTROL-S — suspends sending.
CONTROL-Q — resumes sending.
CONTROL-O — skips to end of file or message.
CONTROL-U — cancels input for current line.
CONTROL-R — redisplays current line.
CONTROL-X — cancels everything typed ahead but unsent.
CONTROL-C — cancel current activity and start over.

If a particular control key is causing the terminal program to take some other action, then the terminal program should be reconfigured to use any of the other available control keys in place of the one required by Delphi.

Modified Command Card for PCM Readers

DELPHI™

BIG P

Get your message across in a bold way with this banner maker

By Michael J. Himowitz

The modern dot-matrix printer is a flexible and useful tool. It's also a great toy. If you're willing to spend some time fooling with it, you can entertain yourself, your family and friends — and learn something about programming in the process.

When I first got my Gemini printer, I fell in love with its block graphics capabilities. Being a newspaperman by trade, I thought it would be fun to use the printer as a typesetter of sorts, using those graphics to produce headline type.

The result was *Bigprint*. The versions here run on any Tandy 1000 or compatible and the Model 100/200 (24K minimum on the Model 100). The program prints title pages and report covers in large block letters of almost infinite variety, depending on the capabilities of your printer and your imagination. It also mixes lines of block letters with lines of standard printing, centering each line on the page.

In normal mode, it produces six block characters per line on a standard nine-inch printer. If you have a 15-inch printer or use condensed (16 or 17 cpi) type, you can get up to 10 characters per line. Lines printed in normal type may have up to 72 characters. The block characters are created on a grid 11 spaces wide and eight spaces deep. Because printed letters are taller than they are wide and because there is space between each line, the *Bigprint* character still has a vertical appearance.

The appearance can be altered in two ways. First, each block character is composed by default of X's and spaces. The program allows you to change the 'X' to any character your printer will produce. This gives the characters an astounding variety of textures. In the Model 1000 version, you may use a different component character for each line of the *Bigprint* message. The Model 100/200 version uses

Mike Himowitz is system editor for the Baltimore Evening Sun, *proprietor of Federal Hill Software and author of numerous programs. He lives in Baltimore with his wife, two children and six computers. He loves them in that order.*

RINT

the same component character throughout, although it could easily be modified.

Second, the height/width ratio may be varied by changing the length of the line feed (if your printer allows this). If you make the line feeds small enough and use a solid block graphics character, you can produce solid type. The program allows you to send these kinds of commands directly to the printer.

From a progamming standpoint, *Bigprint* illustrates the portability of Microsoft BASIC. The original *Bigprint* was written for the TRS-80 Color Computer. The problem with that machine is its 32-character display, which makes it impossible to preview *Bigprint* output on the screen. When I got my first MS-DOS machine with its 80-column display, I saw the opportunity to change that. As a result, the program has a screen display option that you can use as an attractive titling subroutine in other programs.

Because the Color Computer uses a subset of the GW-BASIC used by the Tandy, IBM and compatibles, the conversion was straightforward. The only hassle was making sure that lowercase input on the Tandy 1000 would be treated the same as uppercase input. The Color Computer normally does not use lowercase, so there is a routine to convert lowercase to uppercase.

This version incorporates three major improvements over my first MS-DOS version, which ran on a Sanyo 555. First, there is a "bounce-bar" menu invented by Frank R. Neal of Columbus, Ohio, which I found on the CompuServe Information Service. Second, I have added a routine that allows you to save a *Bigprint* file to disk. The file may then be copied to a printer in DOS with the `Copy` and/or `Print` commands or read into a word processor. Finally, I have added a routine allowing you to set a different component character for each line.

The Model 100/200 version doesn't have the fancy screen preview or as many bells and whistles as the Model 1000 program, but it does the trick. Interestingly enough, the Model 100/200 version runs on the Tandy 1000 with only one modification, the addition of the following line:

TANDY IS DANDY...

...until you want more memory

Your Tandy 1000 is a good machine. A lot of computer for the money.

But you need to add more memory to get the best performance possible. Tandy wants a lot of money for that. Too much money.

Now there's another way to upgrade your memory. The Zuckerboard Expansion Memory card. A high quality, reliable board at a very reasonable price.

So, if you want to get the most out of your computer, you have a choice.

You can go to Tandy, and Tandy will get the most out of you.

Or, you can come to us. We'll fill up your computer without emptying your wallet.

Board Configuration	Tandy Memory Plus	Zuckerboard Expansion Memory
with 256K	$319.95	$109
with 512K	$519.90	$149
Clock Option	$ 99.95*	$ 39
Warranty	90 Days *with mouse	2 Years

MADE IN U.S.A.

INTRODUCTORY OFFER

Send the ZUCKERBOARDS I need for my TANDY 1000 right away.

Here's my check ☐ or money order ☐
MADE PAYABLE TO A.T.D.

____ EM with 256K @ $109 = ____
____ EM with 512K @ $149 = ____

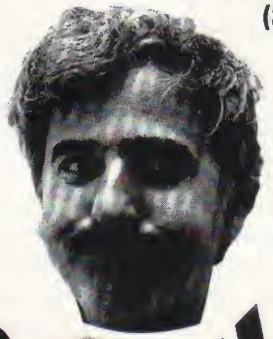
The Doctor pays the shipping.

Subtotal _____
CA residents add 7% tax _____
Amount Enclosed _____

NAME _____
ADDRESS _____
CITY _____ STATE ____ ZIP _____
TELEPHONE ( ) _____

A.T.D. 1287 Lawrence Station Rd., Sunnyvale, CA 94089

CALL TOLL-FREE TO ORDER
CA (800) 458-6200
(800) 624-4920



Dr. Dr. Zucker

It's another ZUCKERBOARD

ATD 1287 Lawrence Station Road • Sunnyvale, CA 94089 • (408) 734-4631

```
585 IF NC=10 THEN WIDTH "LPT1:",132
```

This is a tribute to the consistency of Microsoft BASIC. If you don't feel like typing in the entire Tandy listing, you can use the Model 100/200 version and get good results with a lot less work.

When the program is run, you see a menu with three choices. They are:

```
1. Set Printer Commands
2. Enter Bigprint Message
3. Set Block Character
```

On the Model 1000 version, the cursor keys can be used to position the bounce-bar over your choice and press RETURN, or merely type in the number of your choice.

Option 1 allows you to send a string of commands to the printer. Use this to set condensed type (for 10 characters per line), change the line feed length, or create boldface or italic type. The commands are in the form of ASCII decimal control or escape codes.

Enter one code per line. For example, on my Gemini-10X, the code for condensed type is 15. The codes to reduce the line feed to 1/10 of an inch are 27, 51 and 10. Using *Bigprint*, I type 15 on Command Line 1 and press the RETURN key. On Command Line 2, I type 27 and press RETURN. On Command Line 3, I type 51, and so on.

Make sure your printer is online when you do this because the codes are sent to the printer immediately, producing a single line feed. The paper should be set so the print head is close to the top of a page.

When you are through entering commands, just press the RETURN key with no other input at the Command Line prompt. Your codes are displayed and you are asked if they are correct. If they are, enter Y. Any other response repeats the printer command process.

You are then asked to enter the codes that restore normal printing. This allows you to enter lines of standard type. The procedure is the same as for the first series of commands. The command code sequence for normal type on the Gemini, Epson and similar printers is 27 and 64. Check your printer manual for details. Once you have set the commands, you return to the main menu.

Option 3 allows you to change the character that makes up the *Bigprint* letters. When prompted, just type in the ASCII code of the character desired. The permissible range is 33 to 254. Get these codes from the printer manual, not the computer manual, because the non-alphabetic character sets on all devices are different. The default is the character X, which has an ASCII code of 88. You may not use codes less than 32 (these are reserved for printer control codes).

Option 2 allows you to enter the *Bigprint* message itself. First you are asked whether you want six or 10 characters per line. Remember that standard 10 cpi type on a nine-inch printer yields only six characters per line. To get 10 characters, you must use condensed type or have a 15-inch printer.

Entering each line of the *Bigprint* message requires two operations. If you are using the Model 100/200 version, make sure the CAPS LOCK key is depressed. The first prompt asks whether you want the line to be in standard or *Bigprint* type; enter S or B.

In the Model 1000 version, you may also follow the B with a slash (/) and the ASCII code of the component character you want for that line. For example, if you want the line to be made up of asterisks, you type B/42. The number 42 is the ASCII code for an asterisk. If a slash is not used, you get either an 'X' or the character that is set in Option 3.

The second prompt asks for the message itself. You can use the characters 0 through 9 and A through Z. No punctuation marks or non-alphanumeric characters are allowed. If you want to add them, feel free to modify the program.

Error trapping is fairly solid. You cannot enter a non-alphanumeric character or more than the specified six or 10 characters per line. You may use upper- or lowercase letters on the Model 1000 version. The program converts lowercase to uppercase for output to the printer. On the Model 100/200 version, you must enter all capitals for *Bigprint* lines.

Up to 15 lines in all may be entered. When you are through with the message, press the RETURN key with no other input when prompted for standard or *Bigprint* type. Your message is then displayed. At the beginning of each line are the characters B> or S>. These stand for *Bigprint* or standard type.

If you choose six characters per line, you have the option of a screen display of your message on the Model 1000 version. The program then asks if you want to go to the printer. Enter Y or y to commence printing; any other response results in a prompt asking if you want to save the message to disk. When the message has been printed on paper, you are asked if you want another copy. If you don't, type N. You are asked if you want a disk save. Type Y to save the file to disk or N to return to the main menu.

For those interested in the structure of the program, I have used a number of arrays to set up the characters. The main variables are found in Line 90 of the Model 1000 program. The array G$ consists of 41 elements that reference the DATA statements in lines 1240-1310. These are all the combinations of X's and spaces required to produce the 36 characters in the set.

In turn, these are referenced by the array labeled L, which is two-dimensional. The eight members of each element in this array refer to particular strings of X's and spaces. For example, G$(7) looks like this: " XXX ". That's four spaces, three X's and four spaces. If you look at the DATA statement in Line 1430, you will see eight 7's. This tells the computer to print the string G$(7) eight times. The result is the character 'I'.

You may notice that there are 40 elements in array L and only 36 characters. I left room here for some punctuation marks, but after mapping out and debugging 36 characters, I'd had it. The characters begin with A in Line 1330 and continue sequentially through the alphabet, followed by the numbers.

The Model 1000 program is well-commented. Most of it is standard input and string manipulation. The tricky part is in lines 1710-2000. This is the routine that converts the message, character by character, into *Bigprint* output and centers each line on the page. The corresponding screen output routine begins at Line 2120. You'll see similar language in the Model 100/200 version, although comments have been left out in the interest of conserving memory.

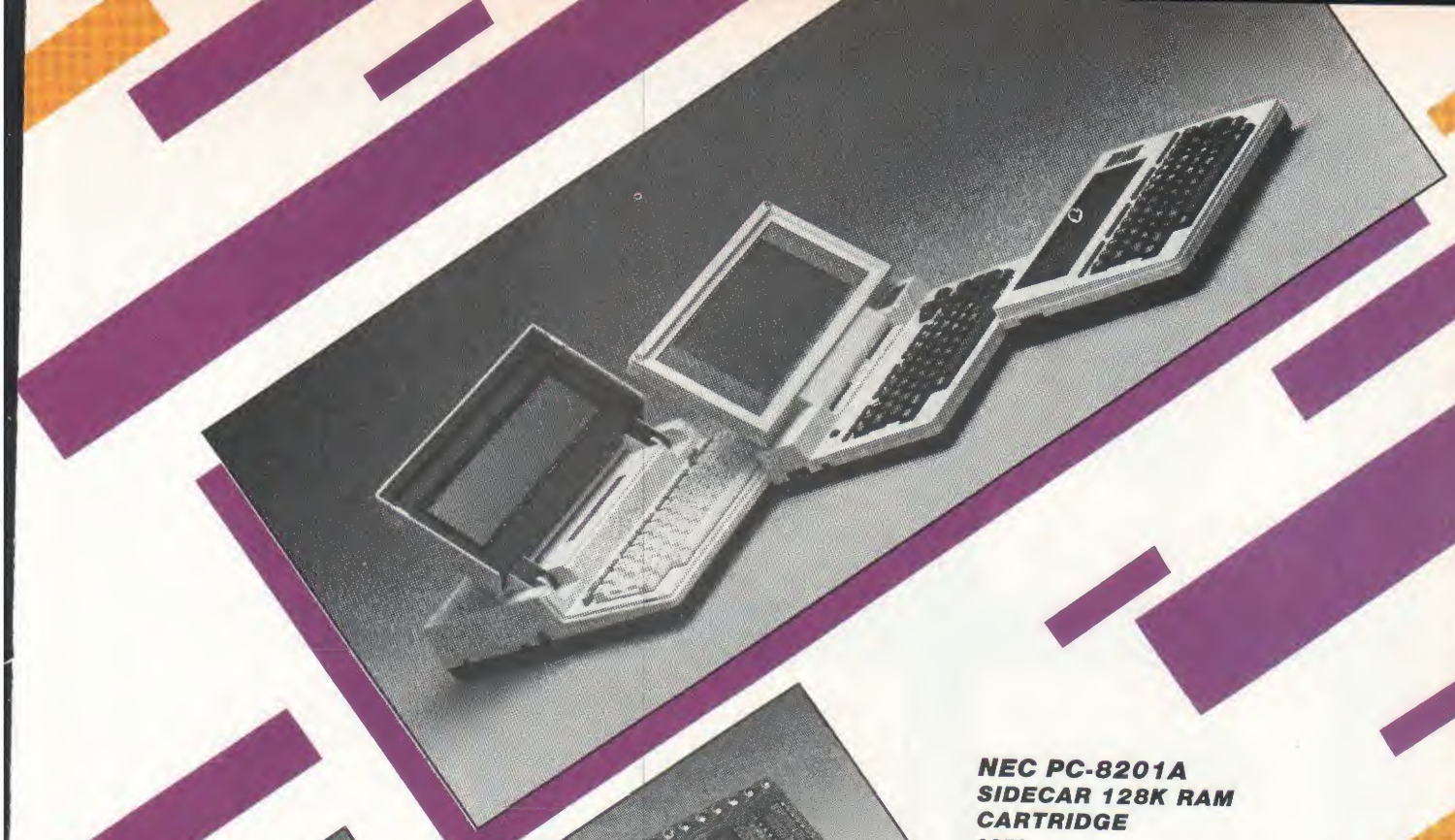
I had a lot of fun working it all out, and I hope you and your family enjoy the program. If you don't want to type your fingers to the bone, send a check for $7 and I'll send you a Tandy 1000 disk or Model 100/200 tape. I can be reached at 8134 Scotts Level Road, Baltimore, MD 21208. My CompuServe number is 71655,1327. □

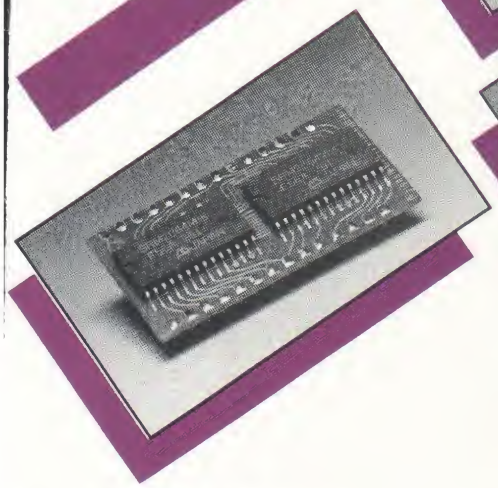
LISTING 1:

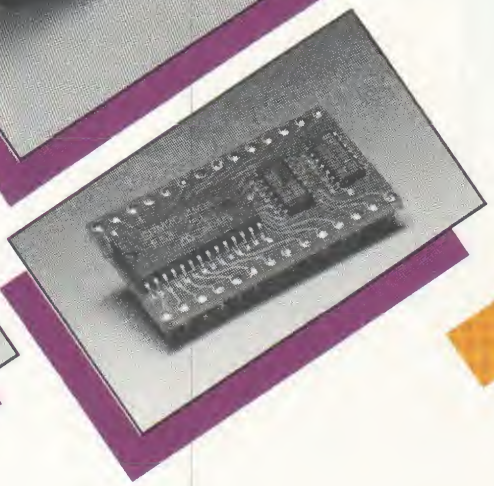
```
1Ø REM ************************************************
2Ø REM ****************** BIGPRINT ********************
3Ø REM ********** BY MICHAEL J. HIMOWITZ **************
4Ø REM ******* TANDY 1ØØØ AND COMPATIBLES **********
5Ø REM ************************************************
6Ø REM
7Ø REM INITIALIZE VARIABLES, READ CHARACTER STRINGS
8Ø KEY OFF
9Ø DIM G$(41), L(4Ø,8), J$(15), M(15), TF$(15), RW(2Ø), RX(2Ø)
1ØØ BE=88:TF(1)=88:TF(2)=42:TF(3)=221
11Ø GOSUB 33ØØ
12Ø CLS:LOCATE 3,5:PRINT"ONE MOMENT PLEASE . . ."
13Ø FOR X=1 TO 41:READ G$(X)
14Ø NEXT X
15Ø FOR X=1 TO 36
16Ø FOR Y=1 TO 8
17Ø READ L(X,Y)
18Ø NEXT Y
19Ø NEXT X
2ØØ RESTORE
21Ø REM
22Ø REM *** BIGPRINT TITLE SCREEN ***
23Ø REM
24Ø GOSUB 254Ø:COLOR FGD,BKGD,BRDR:CLS:LOCATE 2,1:GOSUB 214Ø:FOR DLY=1 TO 4ØØØ:N
EXT:GOTO 335Ø
25Ø REM
26Ø REM *** ENTER BIGPRINT MESSAGE ***
27Ø REM
28Ø GOSUB 61Ø:CLS:LOCATE 3,24:PRINT "ENTER BIGPRINT MESSAGE":PRINT
29Ø PRINT"YOU MAY ENTER UP TO 15 LINES WITH UP TO"NC"CHARACTERS IN EACH LINE.":P
RINT "YOU MAY TERMINATE THE MESSAGE AT ANY TIME BY STRIKING THE <RETURN> KEY"
3ØØ PRINT "AT THE INITIAL LINE PROMPT. ";:INPUT"HIT <RETURN> NOW TO BEGIN.",PE
31Ø CLS:PRINT "USE B/XXX TO CREATE BIGPRINT CHARACTER OF CHR$(XXX)":PRINT:FOR J=
1 TO 15
32Ø PRINT "LINE NO. "J;:INPUT" (S)tandard or (B)igprint";TF$(J):GOSUB 1Ø7Ø:IF TF
$(J)="B" THEN PRINT" ("NC" CHRACTERS MAXIMUM)":PRINT STRING$(NC,".")
33Ø IF TF$(J)="S" THEN PRINT "(72 CHARACTERS MAXIMUM)":PRINT STRING$(72,".")
34Ø IF TF$(J)<>"B" AND TF$(J)<>"S" AND TF$(J)<>"s" AND TF$(J)<>"b"  THEN PRINT"'
B' OR 'S', PLEASE":BEEP:GOTO 32Ø
35Ø LINE INPUT J$(J)
36Ø IF TF$(J)="B" AND LEN(J$(J))> NC THEN BEEP:PRINT"LINE TOO LONG":PRINT"CORREC
TED ";:GOTO 32Ø
37Ø IF TF$(J)="B" THEN GOSUB 26ØØ
38Ø IF BADFLAG=1 THEN BEEP:PRINT "ILLEGAL ENTRY. YOU MUST USE CHARACTERS Ø-9 OR
A-Z.":BADFLAG=Ø:PRINT:PRINT "CORRECTED ";:GOTO 32Ø
39Ø IF TF$="S" AND LEN(J$(J))> 72 THEN BEEP:PRINT "LINE TOO LONG (72 CHARACTERS
MAXIMUM)":PRINT "CORRECTED ";:GOTO 32Ø
4ØØ NEXT J
41Ø J=J-1:GOTO 45Ø
42Ø REM
43Ø REM *** CHECK MESSAGE BEFORE PRINTING ***
44Ø REM
45Ø CLS:PRINT"YOUR MESSAGE:":FOR K=1 TO J:PRINT TF$(K)+"> "+J$(K):NEXT:PRINT
```

```
46Ø REM IF 6 CHARACTER LINE, OFFER SCREEN PREVIEW
47Ø IF NC=6 THEN INPUT; "DO YOU WANT A SCREEN PREVIEW (Y/N)?",SP$:IF SP$="Y" OR
SP$="y" THEN CLS:GOSUB 214Ø:FOR DLY=1 TO 3ØØØ:NEXT:PRINT:INPUT "HIT <RETURN> TO
CONTINUE",PE
48Ø PRINT:LINE INPUT"DO YOU WANT TO GO TO THE PRINTER NOW? (Y/N) ",PN$:IF PN$="Y
" OR PN$="y" THEN 54Ø
49Ø INPUT "DO YOU WANT TO SAVE THIS MESSAGE TO DISK? (Y/N)";DK$:IF DK$<>"Y" AND
DK$<>"y" THEN 115Ø
5ØØ GOTO 285Ø
51Ø REM
52Ø REM ********** EXECUTE BIGPRINT **********
53Ø REM
54Ø INPUT "POSITION PRINT HEAD AT TOP OF PAPER AND HIT <RETURN>",PE
55Ø GOSUB 174Ø:INPUT "ANOTHER COPY (Y/N)";ANOTHER$:IF ANOTHER$="Y" OR ANOTHER$="
y" THEN 54Ø
56Ø INPUT "SAVE THIS FILE TO DISK (Y/N)";SQ$:IF SQ$<>"Y" AND SQ$<>"y" THEN 334Ø
57Ø GOTO 285Ø
58Ø REM
59Ø REM *** SET NUMBER OF CHARACTERS PER LINE ***
6ØØ REM
61Ø CLS:PRINT:PRINT"           BIGPRINT":PRINT
62Ø PRINT"HOW MANY CHARACTERS PER LINE?"
63Ø PRINT:PRINT"THERE IS A 6 CHARACTER LIMIT AT STANDARD 1Ø CPI.":PRINT"1Ø CHARA
CTERS REQIRES CONDENSED FONT OR 132-COLUMN PRINTER.":PRINT:INPUT"YOUR CHOICE (6
OR 1Ø): ";NC
64Ø IF NC<>6 AND NC<>1Ø THEN 61Ø
65Ø IF NC=1Ø THEN WIDTH "LPT1:",132
66Ø RETURN
67Ø REM
68Ø REM *** SET PRINTER COMMANDS ***
69Ø REM
7ØØ GOSUB 274Ø:LOCATE 5,28:PRINT"PRINTER COMMANDS"
71Ø LOCATE 8,6:PRINT "You may enter up to 2Ø direct printer commands. They shoul
d be in the";:LOCATE 9,6:PRINT"form of decimal ASCII control or escape codes. Co
nsult your printer";
72Ø LOCATE 1Ø,6:PRINT "manual for details. You will be prompted for one command
at a time. To end";:LOCATE 11,6:PRINT"the command string, hit the RETURN key wit
h no other input when prompted";
73Ø LOCATE 12,6:PRINT"for the command line."
74Ø LOCATE 15,6:PRINT"DO YOU WANT TO ENTER THESE ";:INPUT"PRINTER COMMANDS (Y/N)
";PC$
75Ø IF PC$<>"Y" AND PC$<>"y" THEN 115Ø
76Ø LOCATE 17,6:INPUT "MAKE SURE PRINTER IS ON LINE--THEN HIT <RETURN>",PE
77Ø CLS:CM=1
78Ø PRINT"COMMAND NO. "+STR$(CM)+":";:INPUT RW(CM):IF RW(CM)=Ø THEN 81Ø
79Ø CM=CM+1
8ØØ GOTO 78Ø
81Ø CLS:PRINT"HERE IS THE STRING OF COMMANDS  YOU HAVE JUST ENTERED":PRINT
82Ø FOR D=1 TO CM-1:PRINT"CHR$("+STR$(RW(D))+") ";:NEXT D
83Ø PRINT:PRINT:INPUT "IS THIS CORRECT (Y/N)";CR$:IF CR$<>"Y" AND CR$<>"y" THEN
PRINT "CORRECTED": GOTO 77Ø
84Ø FOR D=1 TO CM-1:LPRINT CHR$(RW(D));:NEXT D:LPRINT
85Ø CLS:PRINT "INPUT THE COMMANDS THAT RESTORE NORMAL PRINTING.":PRINT "HIT <RET
URN> WITH NO OTHER INPUT WHEN FINISHED."
86Ø NP=1
87Ø PRINT"COMMAND NO."NP":";:INPUT RX(NP):IF RX(NP)=Ø THEN 89Ø
88Ø NP=NP+1:GOTO 87Ø
89Ø CLS:PRINT"THESE ARE THE COMMANDS TO RESTORE NORMAL PRINTING":PRINT
9ØØ FOR E=1 TO NP-1:PRINT"CHR$("RX(E)")";:NEXT E:PRINT
```

```
91Ø PRINT:INPUT "IS THIS CORRECT (Y/N)";CR$:IF CR$="Y" OR CR$="y" THEN 115Ø
92Ø PRINT"corrected":GOTO 86Ø
93Ø REM
94Ø REM *** SET LETTER BLOCK CHARACTER ***
95Ø REM
96Ø CLS:LOCATE 3,1:PRINT "THE CURRENT CHARACTER FOR LETTER BLOCKS IS "+CHR$(34)+
CHR$(BE)+CHR$(34)+", WHICH HAS AN ASCII":PRINT "CODE VALUE OF";:PRINT BE;:PRINT
". ";:INPUT "DO YOU WANT TO SUBSTITUTE ANOTHER CHARACTER (Y/N)";YD$
97Ø IF YD$<>"y" AND YD$<>"Y" THEN BE=88:GOTO 115Ø
98Ø PRINT:INPUT"ENTER THE ASCII CODE FOR THE CHARACTER YOU WANT ";BE
99Ø IF BE < 33 OR BE > 254 THEN BEEP:PRINT "ASCII CODE MUST BE BETWEEN 33 AND 25
4. ";:INPUT "HIT <RETURN> TO CONTINUE",PE:PRINT:GOTO 98Ø
1ØØØ PRINT:PRINT "ONE MOMENT PLEASE, WHILE I CHANGE THE CHARACTER CODE..."
1Ø1Ø REM
1Ø2Ø FOR U=1 TO 41:FOR UU=1 TO LEN(G$(U))
1Ø3Ø IF MID$(G$(U),UU,1)<>CHR$(32) THEN MID$(G$(U),UU,1)=CHR$(BE)
1Ø4Ø LOCATE 2Ø,25:PRINT "NO."U;
1Ø5Ø NEXT UU:NEXT U
1Ø6Ø GOTO 115Ø
1Ø7Ø IF LEN(TF$(J))>1 THEN TF(J)=VAL(MID$(TF$(J),3)) ELSE TF(J)=BE
1Ø8Ø TF$(J)=LEFT$(TF$(J),1)
1Ø9Ø IF TF$(J)="b" THEN TF$(J)="B"
11ØØ IF TF$(J)="s" THEN TF$(J)="S"
111Ø IF TF$(J)="" THEN 41Ø ELSE RETURN
112Ø REM
113Ø REM ********** THE MAIN MENU *********
114Ø REM
115Ø GOTO 334Ø
116Ø CH$=INKEY$:IF CH$="" THEN 116Ø
117Ø ON CH GOTO 12ØØ,28Ø,96Ø,1Ø3Ø
118Ø GOTO 334Ø
119Ø GOTO 115Ø
12ØØ GOSUB 7ØØ:GOTO 115Ø
121Ø REM
122Ø REM THESE DATA STATEMENTS REPRESENT THE COMPONENT PARTS OF THE LETTERS
123Ø REM
124Ø DATA "XXX     XXX",XXXXXXXXXXX,"XXXXXXX    "," XXXXXXXXX ","XXXXXXXXXX "
125Ø DATA XXX   XXXXX,"    XXX    ",XXX  XX XXX,"XXX    XXX ","XXX  XXX   "
126Ø DATA "XXX   XXX  ",XXX    XXXX,XXXX   XXXX,XXX XXX XXX,XXX  X  XXX
127Ø DATA XXXX    XXX,XXXXX   XXX,XXX    XXXX,XXX   XXXXX," XXXXXXX  X"
128Ø DATA " XXX   XXX "," XXX XXX  "," XXXXX   "," X    "," XXXXXXX  "
129Ø DATA "    XXXXXX ", "  XXXX    ", "   XXX    ", " XXXX     ", "   XXXX
   "
13ØØ DATA "  XXXXX    "," XXXXXX "," XXXXXXX","        XXX","XXX        ",
" XXXXXXXXXX"
131Ø DATA  "          ", "   XXXX    ", "XXXXXXXXX  "," XXXXX    ", " XXXXXXXX
XX"
132Ø REM
133Ø REM THESE DATA STATEMENTS REPRESENT THE LETTERS AND NUMBERS
134Ø REM
135Ø DATA 4,2,1,1,2,2,1,1
136Ø DATA 5,2,1,5,2,1,2,5
137Ø DATA 4,2,35,35,35,35,2,4
138Ø DATA 5,2,1,1,1,1,2,5
139Ø DATA 2,2,35,3,3,35,2,2
14ØØ DATA 2,2,35,3,3,35,35,35
141Ø DATA 36,2,35,6,6,1,2,36
142Ø DATA 1,1,1,2,2,1,1,1
143Ø DATA 7,7,7,7,7,7,7,7
```







TANDY 200
24K Memory Module
$89 — two for $170
Can be installed in less than 2 minutes, the only tool required is a coin to pop open the hatch. The TANDY 200 is expandable to 72K by adding two of our 24K modules. Built with the lowest power and most reliable memory chips available. The pins are made specifically for the TANDY 200 chip socket. 2-Year Warranty.

MODEL 100
NEC PC-8201A, Olivetti M10
8K Memory Module
$29 — three for $79
Plugs into the socket provided—expanding your memory by 8,192 bytes. Super low power components for long battery life. Low profile which is very important to properly fit the NEC PC-8201A. Tens of thousands sold since 1983. 2-Year Warranty.

TANDY 600
96K Memory Unit **NEW!**
$249 — two for $469
You can install one or two of these units inside your TANDY 600 — giving you a maximum of 224K.

NEC PC-8201A
SIDECAR 128K RAM CARTRIDGE
$259
Plugs into side expansion slot providing four additional banks of 32K each.

IT'S EASY TO ORDER
Send your order with payment to the address below. Or, if you prefer, credit card orders can be handled by phone—VISA, MasterCard, and American Express are welcome. California residents add 6% sales tax. Checks allow 3 weeks to clear. Institutions and Corporations call for purchase order approval.

1-800-732-5012 TOLL FREE
Orders only (8am. - 5pm. PST)

In California ***(805) 987-4788***
For orders or customer service

If you would like our latest product catalog or to find a dealer near you, please write or call (805) 987-4788

PURPLE COMPUTING
420 Constitution Ave.
Camarillo, CA 93010
Telex: 888661 (PURPLE)

Canada: Canada Portable Computer, Langley, B.C. (604) 856-8858

RAM EXPANSIONS!

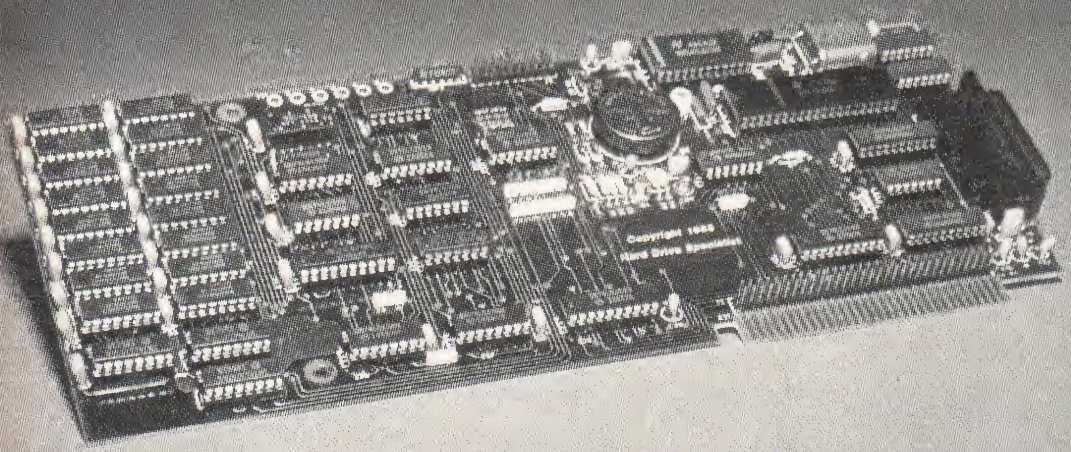
FOOLISH TO PAY MORE — DANGEROUS TO PAY LESS

```
1440 DATA 34,34,34,34,1,1,2,4
1450 DATA 9,11,10,3,3,10,11,12
1460 DATA 35,35,35,35,35,35,2,2
1470 DATA 1,13,2,14,15,1,1,1
1480 DATA 1,16,17,14,19,18,1,1
1490 DATA 4,2,1,1,1,1,2,4
1500 DATA 5,2,1,2,5,35,35,35
1510 DATA 4,2,1,1,8,1,5,20
1520 DATA 5,2,1,5,2,1,1,1
1530 DATA 36,2,35,5,36,34,2,5
1540 DATA 2,2,7,7,7,7,7,7
1550 DATA 1,1,1,1,1,1,2,4
1560 DATA 1,1,1,21,22,23,7,24
1570 DATA 1,1,1,15,14,2,1,1
1580 DATA 1,1,21,23,25,21,1,1
1590 DATA 1,21,22,23,7,7,7,7
1600 DATA 2,2,32,30,27,29,2,2
1610 DATA 37,37,37,37,37,37,37,37
1620 DATA 38,40,7,7,7,7,7,7
1630 DATA 5,2,34,4,39,35,2,2
1640 DATA 5,2,34,32,33,34,2,5
1650 DATA 1,1,1,2,2,34,34,34
1660 DATA 2,2,35,5,2,34,2,5
1670 DATA 35,35,35,5,2,1,2,4
1680 DATA 2,2,34,34,34,34,34,34
1690 DATA 4,2,1,4,2,1,2,4
1700 DATA 4,2,1,2,41,34,2,4
1710 REM
1720 REM *** THIS SENDS BIGPRINT MESSAGE TO THE PRINTER ***
1730 REM
1740 FOR K=1 TO J
1750 IF TF$(K)="S" THEN GOSUB 2030:GOTO 1970
1760 FOR Y=1 TO 8
1770 LL=LEN(J$(K)):TL=LEN(J$(K))*13
1780 IF NC=10 THEN TL=INT((128-TL)/2) ELSE IF NC=6 THEN TL=INT((78-TL)/2)
1790 IF TL<0 THEN TL=1
1800 LPRINT STRING$(TL,32);
1810 FOR V=1 TO LEN(J$(K))
1820 IF MID$(J$(K),V,1)="0" THEN MID$(J$(K),V,1)="O"
1830 Z$(V)=MID$(J$(K),V,1)
1840 IF Z$(V)=CHR$(0) THEN Z$(V)=CHR$(32)
1850 IF V>LEN(J$(K)) THEN 1960
1860 M(V)=ASC(Z$(V))-64
1870 IF M(V)=-32 THEN M(V)=27:GOTO 1890
1880 IF M(V)<-6 AND M(V)>-16 THEN M(V)=M(V)+43
1890 IQ$=G$(L(M(V),Y))+"  "
1900 FOR WR=1 TO LEN(IQ$):IF MID$(IQ$,WR,1)<>CHR$(32) THEN MID$(IQ$,WR,1)=CHR$(T
F(K))
1910 NEXT WR
1920 LPRINT IQ$;
1930 NEXT V
1940 LPRINT
1950 NEXT Y
1960 LPRINT:LPRINT
1970 NEXT K
1980 RETURN
1990 REM
2000 REM
2010 REM ****** STANDARD LPRINTING *****
```

```
2Ø2Ø REM
2Ø3Ø IF TF$(K-1)="S" THEN LPRINT
2Ø4Ø IF TF$(K-1)="B" THEN FOR U=1 TO 2:LPRINT:NEXT U
2Ø5Ø LL=LEN(J$(K))
2Ø6Ø TL=INT((76-LL)/2)
2Ø7Ø FOR E=1 TO NP-1:LPRINT CHR$(RX(E));:NEXT E:LPRINT:LPRINT STRING$(TL,32)+J$(
K)
2Ø8Ø IF TF$(K+1)="S" THEN LPRINT
2Ø9Ø IF TF$(K+1)="B" THEN FOR D=1 TO CM-1:LPRINT CHR$(RW(D));:NEXT D:FOR U=1 TO
3:LPRINT:NEXT U
21ØØ RETURN
211Ø REM
212Ø REM ****** SCREEN BIGPRINT ROUTINE *******
213Ø REM
214Ø FOR K=1 TO J
215Ø IF TF$(K)="S" THEN GOSUB 241Ø:GOTO 236Ø
216Ø FOR Y=1 TO 8
217Ø LL=LEN(J$(K)):TL=LEN(J$(K))*13
218Ø IF NC=1Ø THEN TL=INT((128-TL)/2) ELSE IF NC=6 THEN TL=INT((78-TL)/2)
219Ø PRINT STRING$(TL,32);
22ØØ FOR V=1 TO LEN(J$(K))
221Ø IF MID$(J$(K),V,1)="Ø" THEN MID$(J$(K),V,1)="O"
222Ø Z$(V)=MID$(J$(K),V,1)
223Ø IF Z$(V)=CHR$(Ø) THEN Z$(V)=CHR$(32)
224Ø IF V>LEN(J$(K)) THEN 235Ø
225Ø M(V)=ASC(Z$(V))-64
226Ø IF M(V)=-32 THEN M(V)=27:GOTO 228Ø
227Ø IF M(V)<-6 AND M(V)>-16 THEN M(V)=M(V)+43
228Ø IQ$=G$(L(M(V),Y))+"  "
229Ø FOR WR=1 TO LEN(IQ$):IF MID$(IQ$,WR,1)<>CHR$(32) THEN MID$(IQ$,WR,1)=CHR$(T
F(K))
23ØØ NEXT WR
231Ø PRINT IQ$;
232Ø NEXT V
233Ø PRINT
234Ø NEXT Y
235Ø PRINT:PRINT
236Ø NEXT K
237Ø RETURN
238Ø REM
239Ø REM ****** SCREEN PRINT STANDARD TYPE ******
24ØØ REM
241Ø IF TF$(K-1)="S" THEN PRINT
242Ø IF FLAG=Ø THEN PRINT:GOTO 244Ø
243Ø IF TF$(K-1)="B" THEN FOR U=1 TO 2:PRINT:NEXT U
244Ø LL=LEN(J$(K))
245Ø TL=INT((76-LL)/2)
246Ø PRINT STRING$(TL,32)+J$(K)
247Ø IF FLAG=Ø THEN FLAG=1:GOTO 25ØØ
248Ø IF TF$(K+1)="S" THEN PRINT
249Ø IF TF$(K+1)="B" THEN FOR D=1 TO CM-1:PRINT CHR$(RW(D));:NEXT D:FOR U=1 TO 3
:PRINT:NEXT U
25ØØ RETURN
251Ø REM
252Ø REM *** SET UP TITLE SCREEN MESSAGE ***
253Ø REM
254Ø J$(1)="TANDY":J$(2)="BIG":J$(3)="PRINT":J$(4)="BY MICHAEL J. HIMOWITZ":TF$(
1)="B":TF$(2)="B":TF$(3)="B":TF$(4)="S"
255Ø NC=6:J=4:CLS
```

```
256Ø RETURN
257Ø REM
258Ø REM ********** CHECK LINE FOR PROPER CONTENT ******
259Ø REM
26ØØ FOR CHAR=1 TO LEN(J$(J))
261Ø CHAR$=MID$(J$(J),CHAR,1)
262Ø IF CHAR$=>"A" AND CHAR$<="Z" THEN 267Ø
263Ø IF CHAR$=>"Ø" AND CHAR$<="9" THEN 267Ø
264Ø CODE=ASC(CHAR$)
265Ø GOSUB 269Ø
266Ø IF CODE=>97 OR CODE<=122 THEN MID$(J$(J),CHAR,1)=CHR$(CODE-32)
267Ø NEXT CHAR
268Ø RETURN
269Ø IF CODE=32 THEN RETURN
27ØØ IF CODE<48 THEN GOSUB 275Ø
271Ø IF CODE > 57 AND CODE < 65 THEN GOSUB 275Ø
272Ø IF CODE > 9Ø AND CODE < 96 THEN GOSUB 275Ø
273Ø RETURN
274Ø RETURN
275Ø BADFLAG=1:RETURN
276Ø CLS:BEEP:PRINT:PRINT:PRINT "SORRY, BUT THERE HAS BEEN, AS THEY SAY IN THE T
RADE, A FATAL ERROR":PRINT:INPUT "HIT <RETURN> TO RESTART PROGRAM",PE:RUN
277Ø CLS:LOCATE 3,3Ø:PRINT"BIGPRINT MENU"
278Ø LOCATE 7,1Ø:PRINT "A. SET PRINTER COMMANDS"
279Ø LOCATE 9,1Ø:PRINT "B. ENTER BIGPRINT MESSAGE"
28ØØ LOCATE 11,1Ø:PRINT"C. SET BLOCK CHARACTER"
281Ø RETURN
282Ø REM
283Ø REM ********** PRINT FILE TO DISK **********
284Ø REM
285Ø CLS:LOCATE 2,22:PRINT "SEND BIGPRINT MESSAGE TO DISK FILE"
286Ø LOCATE 1Ø,22:LINE INPUT "ENTER A FILENAME.EXT: ";NF$
287Ø LOCATE 12,15:PRINT "PREPARE DISK DRIVE AND HIT <RETURN> TO SAVE "+NF$;:LINE
 INPUT PE$
288Ø OPEN NF$ FOR OUTPUT AS #1
289Ø GOSUB 327Ø
29ØØ REM
291Ø FOR K=1 TO J
292Ø IF TF$(K)="S" THEN GOSUB 319Ø:GOTO 313Ø
293Ø FOR Y=1 TO 8
294Ø LL=LEN(J$(K)):TL=LEN(J$(K))*13
295Ø IF NC=1Ø THEN TL=INT((128-TL)/2) ELSE IF NC=6 THEN TL=INT((78-TL)/2)
296Ø IF TL>Ø THEN PRINT#1, STRING$(TL,32);
297Ø FOR V=1 TO LEN(J$(K))
298Ø IF MID$(J$(K),V,1)="Ø" THEN MID$(J$(K),V,1)="O"
299Ø Z$(V)=MID$(J$(K),V,1)
3ØØØ IF Z$(V)=CHR$(Ø) THEN Z$(V)=CHR$(32)
3Ø1Ø IF V>LEN(J$(K)) THEN 312Ø
3Ø2Ø M(V)=ASC(Z$(V))-64
3Ø3Ø IF M(V)=-32 THEN M(V)=27:GOTO 3Ø5Ø
3Ø4Ø IF M(V)<-6 AND M(V)>-16 THEN M(V)=M(V)+43
3Ø5Ø IQ$=G$(L(M(V),Y))+"  "
3Ø6Ø FOR WR=1 TO LEN(IQ$):IF MID$(IQ$,WR,1)<>CHR$(32) THEN MID$(IQ$,WR,1)=CHR$(T
F(K))
3Ø7Ø NEXT WR
3Ø8Ø PRINT#1,IQ$;
3Ø9Ø NEXT V
31ØØ PRINT #1,""
311Ø NEXT Y
```

The Ultimate Expansion for the Model 1000!
TanPak ™ by Hard Drive Specialist



7 Functions Include:
DMA
Serial Port
Memory up to 512K
Clock/Calendar
Printer Spooler
Memory Disk
Expansion Port for Future Options

Features Include:
Gold Edge Cards
Pre-Tested and Burned In
Full Documentation
Supporting Software
One Year Warranty
Expandable to 512K

TanPak ™ PCM CERTIFICATION SEAL

The TanPak ™ expansion board has been designed to allow expansion beyond the scope of the standard Model 1000. Seven of the most needed functions/features have been combined into one package using only one expansion slot. Your remaining spaces are left free for future expansion needs.

Seven Function/Features on One Board

512K Memory Expansion

Socketed and expandable to 512K. This is done by two banks of memory using either two 64K increments (128K), or 256K increments (256K or 512K). This allows a total of 640K in the Tandy 1000.

Serial Port

Using the same configuration as the Model 1000 port you are assured of complete compatibility as well as being able to configure it as COM1 or COM2.

Clock-Calendar

Quartz-controlled for a high degree of accuracy, featuring a battery backup.

DMA

The DMA (Direct Memory Access) is used on the Model 1000's first memory card. It increases memory speed and is a must for hard drive operation.

Printer Spooler

Use part of your TanPak ™ memory as a printer buffer. Choose the amount of buffer space you need and stop waiting on your printer.

Memory Disk

Use part of your TanPak ™ memory as a RAM drive. With a solid state drive you can store, retrieve, and sort data quickly and easily.

Expansion System

The TanPak ™ was designed with an expansion port that will allow upgrading with additional features when they become available. Some of the possible features are: a second parallel port, a second serial port, mouse, hard drive port, bubble memory, A/D, and D/A as well as many others.

TanPak ™ 0K . . . . . $329.

Call for pricing on memory modules.

TanPak ™ Secondary

If you already have a Model 1000 memory board and do not wish to replace it, the TanPak ™ Secondary is for you. It features Memory up to 256K, Clock/Calendar, Serial Port, Printer Spooler, and Printer Buffer.

TanPak ™ Secondary 0K . . . . . $249.

TanTel ™

Internal 300/1200 baud Modem for the Tandy 1000. Supported by the DeskMate software package.

TanTel ™ . . . . . $249.

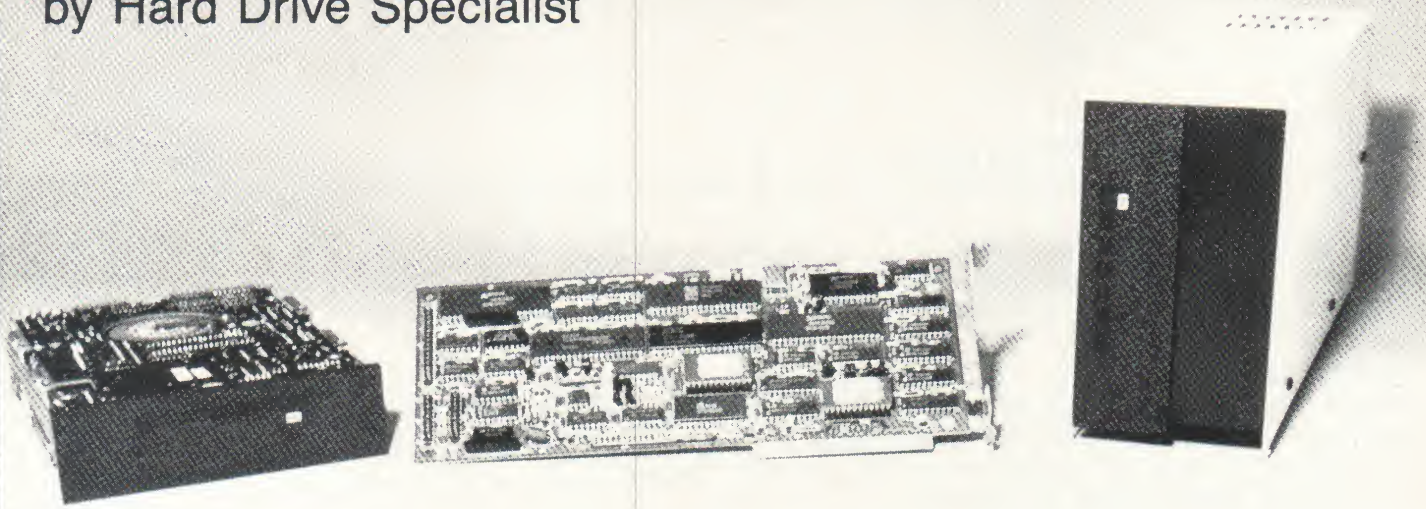
Ordering Information

Use our W.A.T.S. line to place your order via Visa, MasterCard, or Wire Transfer. Or mail your payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request. Tandy and Radio Shack are trademarks of the Tandy Corporation. TanPak, and TanTel are trademarks of Hard Drive Specialist, a division of Compukit Corporation.

Hard Drive Specialist
16208 Hickory Knoll
Houston, Texas 77059
1-713-480-6000
Order Line
1-800-231-6671

Megabytes for Mini-Bucks

Tandy 1000 Hard Drives
by Hard Drive Specialist



Shop and compare. Hard Drive Specialist has been building hard drive systems for years and have sold thousands of subsystems to satisfied Radio Shack/Tandy customers. We use the latest state-of-the-art disk drives and controllers. Our drives all use buffered seek logic and plated media to result in almost one-fourth the average access times found on our competitors drives. Plated media results in longer platter life and high resistance to head crashes not found on coated media drives. We utilize high quality construction throughout including heavy duty power supplies and gold edge card connectors. Internal drive systems include an interface card and a half-height hard drive that replaces the top disk drive in both size and power consumption. External units include an interface card, case, power supply and hard drive unit. All units require a memory board with DMA such as the TanPak™ multifunction card, or a Tandy Board, Part #25-1004.

Hard Disk System Features:

- All sizes rated after formatting.
- One secondary drive may be added.
- Error checking and correcting controller.
- Software drivers included.
- Buffered seek drives for improved access time.
- Built in power up diagnostics.
- Plated media for long disk life.
- Gold connectors used throughout.
- 1 year warranty.
- Boots directly from Hard Drive.
- Uses Tandy 1000 MS DOS.

10 Megabyte Internal	**$ 549.**
20 Megabyte Internal	**$ 749.**
42 Megabyte Internal	**$1349.**
10 Megabyte External	**$ 699.**
20 Megabyte External	**$ 899.**
42 Megabyte External	**$1495.**

Ordering Information

Use our WATS line to place your order via Visa, MasterCard, or Wire Transfer. Or, mail your payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request. Tandy and Radio Shack are trademarks of the Tandy Corporation. TanPak, is a trademark of Hard Drive Specialist, a division of Compukit Corporation.

Hard Drive Specialist
16208 Hickory Knoll
Houston, Texas 77059
1-713-480-6000

Order Line
1-800-231-6671

```
312Ø PRINT#1,"":PRINT #1,""
313Ø NEXT K
314Ø GOTO 328Ø
315Ø REM
316Ø REM
317Ø REM ****** STANDARD PRINTING ******
318Ø REM
319Ø IF TF$(K-1)="S" THEN PRINT#1,""
32ØØ IF TF$(K-1)="B" THEN FOR U=1 TO 2:PRINT#1,"":NEXT U
321Ø LL=LEN(J$(K))
322Ø TL=INT((76-LL)/2)
323Ø FOR E=1 TO NP-1:PRINT#1, CHR$(RX(E));:NEXT E:PRINT#1,"":PRINT#1, STRING$(TL
,32)+J$(K)
324Ø IF TF$(K+1)="S" THEN PRINT#1,""
325Ø IF TF$(K+1)="B" THEN FOR D=1 TO CM-1:PRINT#1,CHR$(RW(D));:NEXT D:FOR U=1 TO
 3:PRINT#1,"":NEXT U
326Ø RETURN
327Ø FOR D=1 TO CM-1:PRINT#1, CHR$(RW(D));:NEXT D:PRINT#1,"":RETURN
328Ø CLOSE#1:LOCATE 14,2Ø:PRINT "FILE "+NF$+" SAVED. HIT <RETURN> FOR MENU.";:LI
NE INPUT PE$:GOTO 334Ø
329Ø RETURN
33ØØ COLOR 7,Ø,Ø:CLS:LOCATE 9,19:PRINT "DO YOU HAVE A COLOR MONITOR? (Y/N)"
331Ø MNR$=INKEY$:IF MNR$="" THEN 331Ø
332Ø IF MNR$<>"Y" AND MNR$<>"y" THEN FGD=7:BKGD=Ø:BRDR=Ø:RETURN
333Ø FGD=7:BKGD=1:BRDR=1:RETURN
334Ø REM *** BAR MENU ROUTINE COURTESY OF FRANK R. NEAL ***
335Ø ROW=8:COL=2Ø:'     SET ROW AND COLUMN FOR MENU
336Ø C1F=FGD:C1B=BKGD '   SET COLOR CODES
337Ø C2F=BKGD:C2B=FGD:'       SET BAR COLOR TO COLOR Ø,2
338Ø M$(1)="SET PRINTER COMMANDS":M$(2)="ENTER BIGPRINT MESSAGE":M$(3)="CHANGE B
LOCK CHARACTER":M$(4)="QUIT THE PROGRAM"
339Ø NP=4:
34ØØ '
341Ø GOSUB 348Ø
342Ø CLS
343Ø ON CH GOTO 12ØØ,28Ø,96Ø,36ØØ
344Ø GOTO 334Ø
345Ø GOTO 341Ø
346Ø '
347Ø '
348Ø COLOR C1F,C1B:CLS:LOCATE ROW,COL+11:PRINT"THE BIGPRINT MENU":LOCATE ROW+1,C
OL:PRINT"Use <ARROWS> to select <ENTER> to Choose":FOR J=1 TO 16:X$=INKEY$:NEXT:
CH=1
349Ø LS=2:FOR J=1 TO NP:IF LEN(M$(J))>LS THEN LS=LEN(M$(J))
35ØØ NEXT:ML$="##  \"+SPACE$(LS-1)+"\":SL=COL+18-LEN(ML$)/2
351Ø FOR K=1 TO NP:LOCATE ROW+2+K,SL:PRINT USING ML$;K,M$(K):NEXT
352Ø LOCATE ROW+2+CH,SL:COLOR C2F,C2B:PRINT USING ML$;CH,M$(CH):COLOR C1F,C1B:TD
=CH
353Ø X$=INKEY$:IF LEN(X$) THEN KP=ASC(RIGHT$(X$,1)) ELSE 353Ø
354Ø IF KP=72 THEN CH=CH-1:IF CH<1 THEN CH=NP
355Ø IF KP=8Ø THEN CH=CH+1:IF CH>NP THEN CH=1
356Ø IF X$=>"1" AND X$<="9" THEN IF VAL(X$)=>1 AND VAL(X$)<=NP THEN CH=VAL(X$):R
ETURN
357Ø IF KP=13 THEN RETURN
358Ø IF KP<>72 AND KP<>8Ø THEN KP=KP-48:IF KP<1 OR KP>NP THEN PRINT CHR$(7):GOTO
 353Ø ELSE CH=KP
359Ø IF CH=TD THEN 353Ø ELSE LOCATE ROW+2+TD,SL:PRINT USING ML$;TD,M$(TD):GOTO 3
52Ø
36ØØ CLS:LOCATE 3,5:PRINT"THANK YOU FOR USING BIGPRINT":END
```

PCM BAR CODED LISTING

LISTING 2:

```
1Ø REM BIGPRINT TANDY MODEL 1ØØ/2ØØ
2Ø REM BY MICHAEL J. HIMOWITZ
3Ø CLEAR 44ØØ
4Ø DIM G$(41), L(4Ø,8), J$(15), M(15), T
F$(1Ø), RW(2Ø), RX(2Ø)
5Ø BE=88
6Ø GOSUB 95Ø
7Ø CLS:PRINT"ONE MOMENT PLEASE....":FOR
X=1 TO 41:READ G$(X)
8Ø IF BE<>88 THEN GOSUB 68Ø
9Ø NEXT X
1ØØ FOR X=1 TO 36
11Ø FOR Y=1 TO 8
12Ø READ L(X,Y)
13Ø NEXT Y
14Ø NEXT X
15Ø RESTORE
16Ø GOSUB 55Ø
17Ø GOTO 18Ø
18Ø CLS:PRINT"YOU MAY INPUT UP TO 15 LIN
ES WITH UP TO"NC "LETTERS EACH"
19Ø PRINT:PRINT "TO BEGIN, ";:INPUT"PRES
S <ENTER>";PE
2ØØ CLS:FOR J=1 TO 15
21Ø PRINT "LINE NO."J;:INPUT"(S)TANDARD
OR (B)IGPRINT";TF$(J):GOSUB 94Ø:IF TF$(J
)="B" THEN PRINT"("NC" CHRS MAX.)"
22Ø IF TF$(J)<>"B" AND TF$(J)<>"S" THEN
PRINT"'B' OR 'S', PLEASE":SOUND 1ØØØ,5:G
OTO 21Ø
23Ø LINE INPUT J$(J)
24Ø IF TF$(J)="B" AND LEN(J$(J))> NC THE
N SOUND 11ØØ,3:PRINT"LINE TOO LONG":PRIN
T"Corrected":GOTO 21Ø
25Ø GOTO 27Ø
26Ø LL=LEN(J$(J)):VB=NC-LL:VC=INT(VB/2)+
1:J$(J)=STRING$(VC,32)+J$(J)
27Ø NEXT J:GOTO 29Ø
28Ø J=J-1:GOTO 29Ø
29Ø CLS:PRINT"YOUR MESSAGE:":FOR K=1 TO
J:PRINT TF$(K)+">"+J$(K)+"/";:NEXT:PRINT
:PRINT"PRINT THIS? (Y/N)";
3ØØ YN$=INKEY$:IF YN$="" THEN 3ØØ ELSE I
F YN$<>"Y" THEN 95Ø
31Ø CLS:PRINT:LINE INPUT "PREPARE PRINTE
R AND HIT <ENTER> TO PRINT";PE$
32Ø FOR K=1 TO J
33Ø IF TF$(K)="S" THEN GOSUB 6ØØ:GOTO 51
Ø
34Ø FOR Y=1 TO 8
35Ø LL=LEN(J$(K)):TL=LEN(J$(K))*13
36Ø IF NC=1Ø THEN TL=INT((128-TL)/2) ELS
E IF NC=6 THEN TL=INT((78-TL)/2)
37Ø IF TL<Ø THEN TL=Ø
38Ø LPRINT STRING$(TL,32);
39Ø FOR V=1 TO LEN(J$(K))
4ØØ IF MID$(J$(K),V,1)="Ø" THEN MID$(J$(
K),V,1)="O"
41Ø Z$(V)=MID$(J$(K),V,1)
42Ø IF V>LEN(J$(K)) THEN 5ØØ
43Ø M(V)=ASC(Z$(V))-64
44Ø IF M(V)=-32 THEN M(V)=27:GOTO 46Ø
45Ø IF M(V)<-6 AND M(V)>-16 THEN M(V)=M(
V)+43
46Ø LPRINT G$(L(M(V),Y))+"  ";
47Ø NEXT V
48Ø LPRINT
49Ø NEXT Y
5ØØ LPRINT:LPRINT
51Ø NEXT K
52Ø PRINT:PRINT "ANOTHER COPY? (Y/N)"
53Ø CP$=INKEY$:IF CP$="" THEN 53Ø ELSE I
F CP$="Y" THEN 31Ø
54Ø GOTO 95Ø
55Ø GOTO 56Ø
56Ø CLS:PRINT"HOW MANY CHARACTERS PER LI
NE?"
57Ø PRINT"6 CHARACTERS (STANDARD 1Ø CPI)
":PRINT"1Ø CHARACTERS (NEEDS CONDENSED":
PRINT"FONT OR 132-COL PRINTER)":PRINT:IN
PUT"YOUR CHOICE (6 OR 1Ø)";NC
58Ø IF NC<>6 AND NC<>1Ø THEN 55Ø
59Ø RETURN
6ØØ IF TF$(K-1)="S" THEN LPRINT
61Ø IF TF$(K-1)="B" THEN FOR U=1 TO 2:LP
RINT:NEXT U
62Ø LL=LEN(J$(K))
63Ø TL=INT((76-LL)/2)
64Ø FOR E=1 TO NP-1:LPRINT CHR$(RX(E));:
NEXT E:LPRINT STRING$(TL,32)+J$(K)
65Ø IF TF$(K+1)="S" THEN LPRINT
66Ø IF TF$(K+1)="B" THEN FOR D=1 TO CM-1
:LPRINT CHR$(RW(D));:NEXT D:FOR U=1 TO 3
:LPRINT:NEXT U
67Ø RETURN
68Ø FOR U=1 TO LEN(G$(X))
69Ø IF MID$(G$(X),U,1)="X" THEN MID$(G$(
```

```
X),U,1)=CHR$(BE)
7ØØ NEXT U:RETURN
71Ø CLS:PRINT"YOU MAY ENTER UP TO 2Ø DIR
ECT PRINTER   COMMANDS. THEY SHOULD   BE
 IN THE FORM  OF ASCII CONTROL OR ESCAPE
 CODES. TO    END THE STRING OF COMMANDS
, HIT <ENTER> WITH NO OTHER INPUT. DO TH
IS? (Y/N)"
72Ø PC$=INKEY$:IF PC$="" THEN 72Ø
73Ø IF PC$<>"Y" AND PC$<>"y" THEN 95Ø
74Ø CLS:CM=1
75Ø PRINT"COMMAND NO. "+STR$(CM)+":";:IN
PUT RW(CM):IF RW(CM)=Ø THEN 78Ø
76Ø CM=CM+1
77Ø GOTO 75Ø
78Ø CLS:PRINT"HERE IS THE STRING OF COMM
ANDS YOU HAVE JUST ENTERED:"
79Ø FOR D=1 TO CM-1:PRINT"CHR$("+STR$(RW
(D))+") ";:NEXT D
8ØØ PRINT:INPUT "IS THIS CORRECT (Y/N)";
CR$:IF CR$<>"Y" THEN PRINT "corrected":
GOTO 74Ø
81Ø FOR D=1 TO CM-1:LPRINT CHR$(RW(D));:
NEXT D:LPRINT
82Ø CLS:PRINT"INPUT THE COMMANDS THAT RE
STORE NORMAL  PRINTING. HIT <ENTER> ALON
E WHEN DONE"
83Ø NP=1
84Ø PRINT"COMMAND NO."NP":";:INPUT RX(NP
):IF RX(NP)=Ø THEN 86Ø
85Ø NP=NP+1:GOTO 84Ø
86Ø CLS:PRINT"THESE ARE THE COMMANDS TO
RESTORE NORMALPRINTING:"
87Ø FOR E=1 TO NP-1:PRINT"CHR$("RX(E)")"
;:NEXT E
88Ø PRINT:INPUT "IS THIS CORRECT (Y/N)";
CR$:IF CR$="Y" THEN 95Ø
89Ø PRINT"corrected":GOTO 83Ø
9ØØ CLS:PRINT"THE DEFAULT CHARACTER FOR
LETTER BLOCKS IS 'X', WHICH IS ASCII (88
). DO YOU WISHTO SUBSTITUTE ANOTHER CHAR
ACTER? (Y/N)?"
91Ø LINE INPUT YD$:IF YD$<>"Y" THEN BE=8
8:GOTO 93Ø
92Ø INPUT"ENTER THE ASCII CODE FOR THE C
HARACTER  YOU WANT";BE
93Ø GOTO 95Ø
94Ø IF TF$(J)="" THEN 28Ø ELSE RETURN
95Ø CLS:PRINT"                BIGPRINT":P
RINT"
96Ø PRINT"      A. SET PRINTER COMMANDS"
:PRINT"      B. ENTER 'BIGPRINT' MESSAGE
":PRINT"      C. SET BLOCK CHARACTER"
97Ø CH$=INKEY$:IF CH$="" THEN 97Ø
98Ø ON INSTR("AaBbCc",CH$) GOTO 1ØØØ,1ØØ
Ø,7Ø,7Ø,9ØØ,9ØØ
99Ø GOTO 95Ø
1ØØØ GOSUB 71Ø:GOTO 95Ø
1Ø1Ø IF K/6<> INT(K/6) THEN RETURN ELSE
PRINT:PRINT "HIT ANY KEY TO CONTINUE";
1Ø2Ø DATA "XXX     XXX",XXXXXXXXXX,"XXX
XXXX    "," XXXXXXXXX ","XXXXXXXXXX "
1Ø3Ø DATA XXX   XXXXX,"    XXX    ",XXX
 XX XXX,"XXX    XXX ","XXX  XXX   "
1Ø4Ø T1$=INKEY$:IF T1$="" THEN 1Ø4Ø
1Ø5Ø DATA "XXX   XXX  ",XXX    XXXX,XXXX
   XXXX,XXX XXX XXX,XXX  X  XXX
1Ø6Ø DATA XXXX    XXX,XXXXX   XXX,XXX
 XXXX,XXX   XXXXX," XXXXXXX  X"
1Ø7Ø RETURN
1Ø8Ø DATA " XXX   XXX "," XXX XXX  "," 
  XXXXX   "," X    "," XXXXXXX  "
1Ø9Ø DATA "     XXXXXX ", "  XXXX    ",
 "   XXX     ", " XXXX      ", "    XXXX
   "
11ØØ DATA "  XXXXX    "," XXXXXX ","
   XXXXXXX","        XXX","XXX        ",
" XXXXXXXXXX"
111Ø DATA  "          ", "   XXXX    ",
 "XXXXXXXXX  "," XXXXX    ", " XXXXXXXX
XX"
112Ø DATA 4,2,1,1,2,2,1,1
113Ø DATA 5,2,1,5,2,1,2,5
114Ø DATA 4,2,35,35,35,35,2,4
115Ø DATA 5,2,1,1,1,1,2,5
116Ø DATA 2,2,35,3,3,35,2,2
117Ø DATA 2,2,35,3,3,35,35,35
118Ø DATA 36,2,35,6,6,1,2,36
119Ø DATA 1,1,1,2,2,1,1,1
12ØØ DATA 7,7,7,7,7,7,7,7
121Ø DATA 34,34,34,34,1,1,2,4
122Ø DATA 9,11,1Ø,3,3,1Ø,11,12
123Ø DATA 35,35,35,35,35,35,2,2
124Ø DATA 1,13,2,14,15,1,1,1
125Ø DATA 1,16,17,14,19,18,1,1
126Ø DATA 4,2,1,1,1,1,2,4
127Ø DATA 5,2,1,2,5,35,35,35
128Ø DATA 4,2,1,1,8,1,5,2Ø
129Ø DATA 5,2,1,5,2,1,1,1
13ØØ DATA 36,2,35,5,36,34,2,5
131Ø DATA 2,2,7,7,7,7,7,7
132Ø DATA 1,1,1,1,1,1,2,4
133Ø DATA 1,1,1,21,22,23,7,24
134Ø DATA 1,1,1,15,14,2,1,1
135Ø DATA 1,1,21,23,25,21,1,1
136Ø DATA 1,21,22,23,7,7,7,7
137Ø DATA 2,2,32,3Ø,27,29,2,2
138Ø DATA 37,37,37,37,37,37,37,37
139Ø DATA 38,4Ø,7,7,7,7,7,7
14ØØ DATA 5,2,34,4,39,35,2,2
141Ø DATA 5,2,34,32,33,34,2,5
142Ø DATA 1,1,1,2,2,34,34,34
143Ø DATA 2,2,35,5,2,34,2,5
144Ø DATA 35,35,35,5,2,1,2,4
145Ø DATA 2,2,34,34,34,34,34,34
146Ø DATA 4,2,1,4,2,1,2,4
147Ø DATA 4,2,1,2,41,34,2,4
```

PCM

Protect confidential files from prying eyes

Security's the Name of the Game

By John B. Harrell, III
PCM Contributing Editor

This month, I would like to continue with the concept of device drivers, which I touched on briefly when we examined the ANSI.SYS extended screen and keyboard control driver. Device drivers are a rich and fertile area, and also one of the worst explained features of the MS-DOS operating system.

MS-DOS has a wealthy heritage. It is derived directly from that venerable old faithful — CP/M. If you closely examine the functions available, you will note that many of them are identical to their CP/M counterparts. Unfortunately, MS-DOS also suffers from many of the same weaknesses as its predecessor

Keep those Prying Eyes Out

One weakness is particularly annoying — lack of system security. If you are familiar with the Model-III/4 TRS-80 disk operating system, you will probably miss the password protection afforded by this system. Files can be hidden just as in the MS-DOS system, but they can also be assigned two different types of password protection and user protection "levels," which are used to control file access.

Since MS-DOS lacks this type of protection in any form, anyone may walk up to your computer and use it. If

---

John B. Harrell, III has written for microcomputer magazines for three years. He holds a bachelor's degree in computer science and is a software technical expert for Navy electronic support measures systems.

you have a hard disk, this can be particularly annoying. Data with any amount of sensitive information simply cannot be maintained on the hard disk, forcing you to use those slow floppy disks for storage.

My first attempt to remedy this problem was to write a short program in C that read a password from the user and validated it. If the password was valid, the system would proceed. If it was invalid, the system would "hang up." Placed in the AUTOEXEC.BAT file or any other batch file, this effectively prevented inexperienced users from gaining access to my prized disk files.

Alas, MS-DOS is too kind. If you press the CTRL-C or CTRL-BREAK keys at the "right" time during the start-up sequence, COMMAND.COM allows you to terminate the AUTOEXEC batch file. Smart users soon find out that they can get around this type of password protection very effectively.

I had to find another way. It seems that the only time you are isolated from the actions described is before the system has finished starting itself. But how can we accomplish getting our program loaded before the system is ready? Enter the device driver!

After the system completes its diagnostic self-checks, the Boot ROM attempts to load the first sector from the hard disk or, failing this, from the floppy disk in Drive A:. The boot sector loads the BIOS code from the disk, then executes the initialization phase of the BIOS code after it has successfully loaded.

The BIOS code loads the MS-DOS operating system from the disk and initializes it. MS-DOS then loads COMMAND.COM from the disk and the command processor performs several functions. One of the first is to analyze the contents of the CONFIG.SYS file (if present) and perform the specified actions. This is done before the system is ready for execution to allow you to place additional peripheral drivers such as extended memory boards and alternative console devices (ANSI.SYS) in the system before getting control of the system for your applications.

Here's where we will install the password protection scheme. At least we will be able to control who boots up on the system using our disk. Of course, if they bring their own MS-DOS boot disk, it will not work. But, it is better than nothing.

The Mystery of Device Drivers

Never before has one facet of any system been so shrouded in secrecy. While the documentation is technically quite complete, the lack of programming examples hinders your creative ability beyond belief. PC-DOS does contain an excellent example of a device driver, but it is for a RAM disk (block device) and there is nothing to illustrate how to write a character-oriented device.

Listing 1 was written to make use of a reserved feature of the MS-DOS operating system to explore the device driver chain. The operating system maintains a linked list of all the device drivers installed in the system and, once you find the head, following the list is easy. Running Listing 1 produces a list of device driver information similar to the one in Figure 1 (all addresses *may* be different, this one is for my IBM PC XT using PC-DOS 3.0 and my configuration).

The MS-DOS Programmer's Reference Manual states that the NUL device is always the beginning of the list and you can never reassign it. In the system reserved bytes of the *file control block*, or FCB, MS-DOS conveniently places

Figure 1

Device Driver Chain for a Tandy 2000

Notes

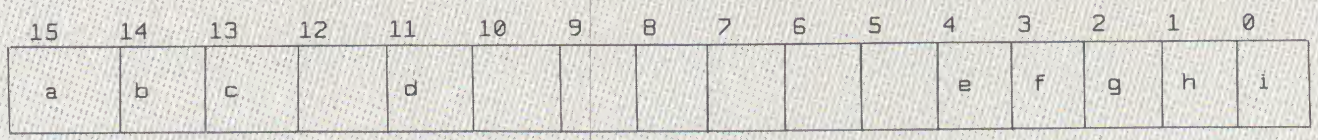
The underlined device entries are installed using the device command in the CONFIG.SYS file. The first entry is a RAM disk driver, the second is the driver for ANSI.SYS (it replaces the functions of the CON driver for Stdin and Stdout), and the third is my password, driver.

Device Driver
Resident Header Chain

Starting Address	Next Hdr Addr	Attr	Strategy Entry Pnt	Interrupt Entry Pnt	Device Name
0127:0154	0892:0000	8004	0127:15A2	0127:15A8	NUL
0892:0000	082F:0000	0800	0892:00A9	0892:00D4	1 Block Device Units
082F:0000	07ED:0000	8013	082F:00A2	082F:00AD	CON
07ED:0000	0070:015D	8000	07ED:0036	07ED:0041	PASSWD$
0070:015D	0070:01EE	8013	0070:00AE	0070:00B9	CON
0070:01EE	0070:029D	8000	0070:00AE	0070:00BF	AUX
0070:029D	0070:0317	8800	0070:00AE	0070:00CE	PRN
0070:0317	0070:03E3	8008	0070:00AE	0070:00E3	CLOCK$
0070:03E3	0070:0200	0800	0070:00AE	0070:00E9	5 Block Device Units
0070:0200	0070:02AF	8000	0070:00AE	0070:00BF	COM1
0070:02AF	0070:0AAB	8800	0070:00AE	0070:00CE	LPT1
0070:0AAB	0070:0ABD	8800	0070:00AE	0070:00D4	LPT2
0070:0ABD	0070:0ACF	8800	0070:00AE	0070:00DA	LPT3
0070:0ACF	0070:FFFF	8000	0070:00AE	0070:00C5	COM2

Figure 1 continued

ATTRIBUTE WORD BIT DEFINITIONS



a. Block/Character Device
b. Device accepts I/O control strings
c. Block device is IBM format
d. Open/Close/Removable Media (DOS 3.x)
e. Device uses special handling (INT 29H)
f. CLOCK Device
g. NUL Device
h. STDOUT Device
i. STDIN Device

the segment and offset addresses for any device you can successfully open. The first section of Listing 1 documents the structure of the FCB, including these reserved areas.

Note that there is a distinction between versions of the MS-DOS operating system. If you are a Tandy 3000 user or are operating your 1000 or 1200 under PC/MS-DOS Version 3.x, then you should delete the comment braces {} around the code for MS-DOS V3.x and comment out or delete the Version 2.x dependent code. Listing 1 will not work correctly on the wrong version.

The next structure documents the format of the device driver header consisting of 18 bytes of information. The first four bytes consist of the segment/offset address pointing to the next device header. The offset address contains $FFFF if the header is the last one in the chain — see the second column of Figure 1. This segment/offset address is also the starting address (first column) of the next device driver in the chain.

The next two-byte field contains the device driver attributes, explaining to the system how the driver is to be used. The breakdown of this field is also contained in Figure 1. The most significant characteristic is the leading bit of the attribute word denoting the two broad classes of device drivers — character and block device. We will primarily be interested in the character devices for this discussion of password protection.

After initializing the display, the "device chain" program opens the NUL device, suing the standard Version 1.x MS-DOS file open request. This returns the updated FCB for our access — remember, this now contains the start of the chain for the device drivers. The next major section of Listing 1 uses this information to "walk" the device header chain and prints the information on the video display.

Now back to Figure 1 for a few more words on explanation. This figure was created on my IBM PC XT and several special features are worth noting. First of all, the program reports that I have a total of six block devices installed. My current configuration supports one real floppy disk, three virtual floppy disks (a la SYSTEM/36) and a rigid disk assembly.

The BIOS creates these five entries as one device header, which is determined by the configuration sense-switches on startup. I also have a RAM disk driver installed, bringing the total of the block devices to six.

The third entry in the list is a redefinition of the CON (console) device. This is for the ANSI.SYS driver. Because MS/PC-DOS scans this device list in the same order that we have printed it, you will always find the most recent definition of the device, allowing you to change system characteristics at will.

Installing Password Protection

The first step in installing password protection on your system is to key in the program from Listing 2 and run it. This creates the device driver file PASS.SYS containing the password code with the initial start-up password of password1. This initial password is very important — do not forget it. I'll show you how to change it shortly.

The next step for installation is to copy the new PASS.SYS file to the root directory of your boot disk. Hard disk users may want to create a test floppy disk to try out the password routine prior to "casting stones" at the hard disk.

You now have to add a command to your CONFIG.SYS file. What? You do not have a CONFIG.SYS file? Shame on you. Even floppy disk users can benefit from the commands available for system configuration (but that's another story). If you have already created one for installing ANSI.SYS, then all you need to do is add the following line to the file:

```
DEVICE=PASS.SYS
```

by using *EDLIN* or any other text processor. If PASS.SYS has been installed on your hard disk in a subdirectory, don't forget to specify the pathname in the "device" command above.

Now you are ready for the acid test! Reset the computer and try it out. The very first thing you should see after the copyright notice is the initial message requesting your password. Enter pass word1 as the initial password. Press the ENTER key and you should see the remainder of your boot sequence just as always.

What Happens Next

A program that stays resident and "eats" up memory (however small) without some additional benefit cannot be too worthwhile. You are right! There is more to PASS.SYS than meets the eye on startup. As part of its initialization process, it has used one of the interrupts that DOS provides for users (INT 67H).

This interrupt allows you to access another part of PASS.SYS from within any program and provides you with an additional security measure. Look at Figure 2 — it is a simple dialogue with DEBUG that creates a stand-alone program called PWRD.COM, which uses this interrupt.

PWRD.COM is not very fancy, but it is

Figure 2
Debug Sequence for Creating PWRD.COM

This file provides a sequence of commands and entries for DEBUG, which creates the file PWRD.COM. PWRD invokes the interrupt 67H capabilities of the PASS.SYS file and allows you to use the password routine from any batch file.

Enter the file exactly as below, typing the underlined parts of the dialogue in response to the computer. Make sure to end each of the lines by pressing the ENTER (RETURN) key. *Do not* type the comments as DEBUG will flag them in error.

```
B:\> debug pwrd.com
File not found
-a100
XXXX:0100 int 67;Invoke password routine at interrupt 67H
XXXX:0102 or ax,ax;Set flags on return result for test
XXXX:0104 jnz 108;Non-zero is an error condition
XXXX:0106 int 20;Return to operating system
XXXX:0108 mov ah,9;Function call to display string
XXXX:010A mov dx,120;Address offset of string
XXXX:010D int 21;Call MS-DOS to display string
XXXX:010F mov ah,7;Function call for Direct Console Input
XXXX:0111 int 21;Call MS-DOS to get a key with no echo
XXXX:0113 jmp 100;Repeat password scan
XXXX:0115;Press ENTER here to exit assembler
-a120;Set message area
XXXX:0120 db 7,7,7,d,a,a,a
XXXX:0127 db 'You have entered the wrong password for access.',d,a,a
XXXX:0159 db 'Press any key to try again...',7,7,'$'
XXXX:0179;Press ENTER here to exit assembler
-rcx;Set CX register with proper byte count
CX 0000
:80
-w;Write PWRD.COM to disk
Writing 0080 bytes
-q;Quit DEBUG and return to DOS

B:\>
```

effective. You can include this one in a batch file and it will not terminate until the correct password has been entered. First of all, it uses INT 67H to interrogate the user via PASS.SYS for the next password (in this case, password2). The password validation flag is returned via the AX register and the next instruction determines its value. If it is zero, the password is good and we return to DOS. If it is non-zero, the program displays an error message and waits for the user to press any key.

After pressing a key, the program loops back for another password attempt and keeps on doing this until you have entered the correct one. If you have my MENU source code, it is a relatively simple feat to add a small code block to it just prior to exiting to the DOS command level. This prevents someone from exiting to DOS unless they have been granted the proper authority.

Because simplicity was the key with this program, I used DOS Function Code 9 to display a string in the video screen. During the brief moment while the string is being displayed, you are susceptible to the CTRL-C interrupt. Rewriting this routine to use the BIOS code display characters routine will alleviate this problem.

One word of caution is appropriate here. I discovered a large difference in the interrupt vector area between the Tandy Model 2000 and the IBM PC. I developed PASS.SYS on the Tandy 2000 and was going to be polite. I would first invoke the previous INT 67H processor to allow it to perform the desired function and would then process the password.

On the Tandy 2000, this would work fine even if no other program had used this vector. The BIOS initialization stores the address of an immediate return instruction here and you effectively get a NOP. When I first tried it on the IBM PC, it took an immediate and lengthy "lunch break," forcing me to turn it off to restore it. I really clobbered something.

I figured PASS.SYS was conflicting with one of the myriad number of resident programs I usually have installed, so I removed them. It immediately took a second lunch break — definitely time to trouble-shoot. It turns out the IBM PC does not initialize this area and leaves it set to all zeros. Oh well, the moral of this story is *do not use* PWRD.COM or any other INT 67H access if PASS.SYS is not installed.

Assigning Your Own Passwords

Figure 3 contains yet another dialogue for DEBUG that allows you to assign your own passwords to PASS.SYS. The first password assigned (offset 02F5H) is used for initial system load and the second password assigned is used for subsequent calls to PASS.SYS using PWRD or INT 67H.

Assign your own passwords where I have indicated "first" and "second password" in the dialogue. There are two important considerations that must be followed for PASS.SYS to work correctly: make sure your assigned password does not exceed 15 characters, and make sure any remainding (unused) characters are filled out with spaces.

Passed Out

I have really found this simple enhancement a real security addition in a situation where anyone can delete your most prized four-megabyte database right before your eyes. Those people who require access and have the proper knowledge can easily be controlled using this scheme.

It certainly is not a panacea for all security problems. File encryption and file locking would do nicely to complement this feature. Until these features are built into the operating system, the only sure security is to lock up your machine or hard disk at night . . . until next month. □

Figure 3

Debug Command Sequence for Changing the Passwords

This dialogue using DEBUG shows how to alter either one or both of the passwords contained in PASS.SYS. Type the portion of the dialogue that is underlined and you will change the passwords from the current default values of password1 and password2 to those indicated below.

It is important to remember that the maximum length of the password string is 15 characters. Do not store any more than this to prevent damaging the remainder of the program. Also, if you are storing a shorter string in the place of a longer one, do not forget to add enough blank characters to the end of the modify steps to fill out the remainder of the strings.

```
B:\pcm> DEBUG PASS.SYS
-D 2F5 L F
1EC5:Ø2FØ                   7Ø 61 73-73 77 6F 72 64 31 2Ø 2Ø        password1
1EC5:Ø3ØØ  2Ø 2Ø 2Ø 2Ø
-E 2F5 'first password '
-D 3Ø4 L F
1EC5:Ø3ØØ                7Ø 61 73 73-77 6F 72 64 32 2Ø 2Ø 2Ø       password2
1EC5:Ø31Ø  2Ø 2Ø 2Ø
-E 3Ø4 'second password'
-D 2F5 L 1E
1EC5:Ø2FØ                   66 69 72-73 74 2Ø 7Ø 61 73 73 77        first passw
1EC5:Ø3ØØ  6F 72 64 2Ø 73 65 63 6F-6E 64 2Ø 7Ø 61 73 73 77   ord second passw
1EC5:Ø31Ø  6F 72 64                                          ord
-W
Writing Ø425 bytes
-Q

B:\pcm>
```

Listing 1: Turbo PASCAL Device Driver Display Program

```
Program Device_Chain;

type

  {=============================================================================
   This data type defines the structure of the PC-DOS/MS-DOS File Control
   Block and is used by this program to access the pointer to the NUL
   device which heads the chain of device driver entries.

   Most of the record is illustrated for documentation and completeness.
   The actual part that is used to access the device driver chain is listed
   in the documentation as "reserved".  Note (in the comments below) that
   the actual location of this pointer in the FCB is different under versions
   2.x and 3.x of the DOS operating system.
  }

  FileControlBlock =
    Record
      Drive      : Byte;
      Filename   : Array [1..8] of char;
      Extension  : Array [1..3] of char;
      CurrentBl  : Integer;
      LRL        : Integer;
      FilSizeLo  : Integer;
      FilSizeHi  : Integer;
      FileDate   : Integer;
      FileTime   : Integer;
{-------------------------------------------------------------------------------
  For DOS 3.Ø and later, the FCB Structure should be as follows:
  remove the alternate braces from the following code segment and
  delete the similar code segment which follows it.
}
{-----------------------   MS/PC-DOS Version 3.x   ---------------------------}
{
      dummy1     : Integer;
      DevOffset  : Integer;
      DevSegment: Integer;
      dummy2     : Byte;
      dummy3     : Byte;
```

PERRY COMPUTERS

CALL
1-800-248-3823

COLOR COMPUTERS	LIST PRICE	OUR PRICE
26-3127 64K Extended Color Computer 2	$199.95	$169.00
26-3134 16K Standard Color Computer 2	119.95	99.00
26-3131 Thinline Disk Drive 0 for Color Computer	299.95	240.00
26-3018 Extended Basic Kit	39.95	36.00
26-3030 OS-9 With Editor Assembler	69.95	59.50
26-3012 Deluxe Joystick	29.95	25.00
26-1208 CCR-81 Recorder	59.95	50.00
30001170 VIP Integrated Library	149.95	139.00
30001110 VIP Writer	69.95	59.00
30001210 Telewriter 64 Tape	49.95	42.00
30001220 Telewriter 64 Disk	59.95	49.00
30009110 Botek Interface		59.00
TANDY 3000		
25-4001 Tandy 3000 One Disk 512K	$2599.00	$1900.00
25-4010 Tandy 3000 One Disk & 20 Meg HD 512K	3599.00	2600.00
25-3046 Deluxe Text Display Adapter	249.95	205.00
26-3047 Deluxe Graphics Display Adapter	499.95	395.00
25-4030 512K to 2 Meg Expansion Board	499.00	395.00
25-4033 Math Co-Processor for 3000	399.95	339.00
25-4050 1.2 Meg Disk Drive Kit for 3000	299.95	255.00
25-4060 Hard Disk Controller Board for 3000	499.00	425.00
25-4062 20 Meg Hard Disk Kit	799.00	679.00
25-4101 MS-DOS 3.1/Basic/DeskMate	99.95	85.00
26-5111 VM-1 Monochrome Monitor	199.95	165.00
26-5112 CM-1 Color Monitor	599.95	510.00
TANDY 100, 200, 600		
26-3901 Tandy 600 Port Comp 32K 3½ DD	$1599.00	$1195.00
26-3904 Basic Rom for Tandy 600	129.95	110.00
26-3910 92K RAM Upgrade for Tandy 600	399.95	340.00
26-3860 Tandy 200 Port Comp 24K	799.00	650.00
26-3866 24K RAM Memory Exp Chip for 200	199.95	170.00
26-3802 Tandy 100 Port Comp 24K	499.00	425.00
26-3816 8K RAM Upgrade for Tandy 100	49.95	42.00
26-3808 Port 3½ Disk Drive for T100 T200	199.95	169.00
26-3815 Portable Disk Case	24.95	21.00
26-3805 Acoustic Coupler	39.95	34.00
26-1409 Printer Cable	14.95	12.70
26-1410 Modem Cable	19.95	17.00
MODEL 4D		
26-1070 Model 4D 64K 2 FD w/DeskMate	$1199.00	$895.00
26-1122 64K Memory Expansion	69.95	65.00
26-1123 Model 3 to Model 4 Upgrade Kit	399.00	385.00
26-1134 Model 4 HD Kit for 15 Meg	79.95	67.95
26-1530 Multiplan Model 4	199.00	169.00
26-1595 Super Scripsit Model 4	199.95	169.00
26-1608 DeskMate for Model 4	195.00	175.00
26-1635 Profile 4 Model 4	249.95	200.00
26-2216 CP/M Plus Model 4	149.00	127.00
26-2231 Double Duty Utility	69.95	58.00
MODEL 6000 COMPUTERS		
26-6021 Model 6000 512K 2 FD Comp	$4499.00	$3125.00
26-6022 Model 6000 512K 1 FD 15 Meg HD	5499.00	3795.00
26-6019 Model 6000 256K Memory Kit	199.95	185.00
26-6052 DT-100 Data Terminal	795.00	600.00
26-1245 10 Meg Disk Cartridge System	2195.00	1750.00
26-6018 6000 Interface Kit for 26-1245	119.95	102.00
25-3022 1000/1200/3000/Interface Kit for 26-1245	199.95	160.00
26-4155 15 Meg Hard Drive Primary	1595.00	1355.00
26-4171 35 Meg Hard Drive Primary	2995.00	2545.00
26-4173 70 Meg Hard Drive Primary	4295.00	3350.00
26-4157 Installation Kit for Primary HD	349.00	295.00

All prices and offers may be changed or withdrawn without notice. Advertised prices are cash prices. C.O.D. accepted ($10.00 charge per carton on C.O.D. Call for further C.O.D. information.) M.C., Visa, add 2%. AX, add 3%. All non-defective items returned will be subject to 10% restocking fee. Defective items require return merchandise authorization. Call for R.M.A. Number before returning. Delivery is subject to product availability.

RADIO SHACK PRINTERS	LIST PRICE	OUR PRICE
26-1276 DMP-105 Dot Matrix Printer	$ 199.00	$ 169.00
26-1275 TRP-100 Portable Thermal Printer	299.00	230.00
26-1280 DMP-130 100 cps Triple Mode Printer	349.95	285.00
26-1268 CGP-220 Color Ink-Jet Printer	699.00	595.00
26-2830 PC-695 8-Color Digital Plotter	799.00	680.00
26-1277 DMP-430 24 Wire Matrix Printer	899.00	635.00
26-1270 DWP-510 43 cps Daisy Wheel	1495.00	1225.00
26-1274 DMP-2100P 24 Dot Matrix Printer	1495.00	1195.00
26-1279 DMP-2200 HiSpeed Matrix Printer	1695.00	1440.00
26-1269 PT-64 Printer Controller	249.95	210.00
26-1498 SW-302 Printer Switch	119.95	100.00
26-1447 Bi-Directional Tractor for DWP 510	219.95	186.95
TANDY 1200, 2000		
25-3000 Tandy 1200 One FD & 10 Meg HD	$1999.95	$1525.00
25-3001 Tandy 1200 Two FD 256K	1499.00	1200.00
25-3010 VM-3 Green Monitor	219.95	185.00
25-1022 CM-10 Color Monitor	459.95	380.00
25-3043 Graphics Display Adapter	299.00	185.00
25-3044 Graphics Master	695.95	540.00
25-3061 Captain Multifunction Board	599.95	475.00
25-3020 TCS-100 Tape Cartridge System	1999.00	1555.00
25-3021 TCS-100 Interface Kit 1000/1200	149.95	120.00
25-3130 MSDOS/BASIC	89.95	76.50
26-5103 Tandy 2000 Two Disk	1599.00	1225.00
26-5104 Tandy 2000 One Disk & 10 Meg HD	2499.00	1885.00
TANDY 1000		
25-1000 Model 1000 128K 1 FD & DeskMate	$ 999.95	$ 705.00
25-1001 Model 1000 256K 1 FD 10 Meg HD	1999.00	1475.00
25-1003 300-Baud Modem Board 1000/1200	149.95	125.00
25-1005 Disk Drive Expansion 1000	199.95	170.00
25-1006 RS-232C Interface 1000/1200	99.95	85.00
25-1013 1200-Baud Modem Board 1000/1200	299.95	230.00
25-1025 10 Meg Hard Disk Drive	699.95	559.00
25-1007 Hard Disk Control Board	299.95	255.00
25-1021 CM-4 Color Monitor	299.95	225.00
25-1022 CM-10 Color Monitor	459.95	380.00
26-3211 VM-2 Monochrome Monitor	149.95	125.00
30051030 PJB Multifunction Board 512K		310.00
EPSON PRINTERS		
20001015 Homewriter 10	$288.00	$215.00
20001025 LX-80 Dot Matrix Printer	299.00	225.00
20001035 FX-85 Dot Matrix Printer	499.00	345.00
20001050 HI-80 4 Pin Plotter	599.00	390.00
20002010 RX-100 Dot Matrix Printer	499.00	400.00
20002030 FX-286 Dot Matrix Printer	749.00	510.00
20003020 LQ-800 24 Pin Head Dot Matrix	799.00	565.00
20003030 LQ-1000 24 Pin Head Dot Matrix	995.00	695.00
10081010 DX-10 Daisy Wheel Printer	299.00	255.00
10081020 DX-20 Daisy Wheel Printer	459.00	390.00
10081030 DX-35 Daisy Wheel Printer	899.00	765.00

For Technical Questions and Information on our complete line of computer accessories and current prices:

CALL 1-517-625-4161
FOR ORDERS ONLY CALL 1-800-248-3823

Mon., Wed. & Fri. 9-9, Tues. & Thurs. 9-6, Sat. 9-3

124 S. MAIN ST. PERRY, MICH. 48872

```
  )
  {----------------------   MS/PC-DOS Version 2.x   ----------------------------}
        dummy1     : Byte;
        DevOffset  : Integer;
        DevSegment: Integer;
        dummy2     : Integer;
        dummy3     : Byte;
  {-------------------------------------------------------------------------------}

        CurRecord : Byte;
        RelRecLo  : Integer;
        RelRecHi  : Integer;
      End;
  {===============================================================================}

    DeviceHeader =
      Record
        NextHeaderOffset : Integer;   { Offset address of next device in chain  }
        NextHeaderSegment: Integer;   { Segment address of next device in chain }
        Attributes       : Integer;   { Device attributes                        }
        StrategyEntPt    : Integer;   { Offset w/i current segment - stragegy   }
        InterruptEntPt   : Integer;   { Offset w/i current segment - interrupt  }
        DeviceName       : Array [1..8] of char;  { Name of the device           }
      End;

    Registers =
      Record
        AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags : Integer;
      End;


    Str8Ø = String[8Ø];


  var
    DeviceControlBlock : FileControlBlock;  { File Control Block for NUL Device }
    Regs               : Registers;         { Machine registers for MS-DOS calls}
    DevicePtr          : ^DeviceHeader;     { Pointer to the next device header }
    DeviceSegment      : integer;           { Track the device segment and      }
    DeviceOffset       : integer;           {    offset while manipulating chain}


  {-------------------------------------------------------------------------------
    HexStr: Converts the integral value passed by the parameter "number" to a
            string of four hexidecimal character digits.
  -------------------------------------------------------------------------------}

  function HexStr ( number : integer ) : Str8Ø;
  const
    HexChars : Array[Ø..15] of char = 'Ø123456789ABCDEF';
  var
    i        : Integer;
    temp     : Str8Ø;
  begin  { function HexStr }
    temp[Ø] := #4;
    for i := 1 to 4 do
  begin
    temp[5-i] := HexChars[ (number and $ØØØF) ];
    number := number shr 4;
  end;
 HexStr := temp;
end;  { function HexStr }


{-------------------------------------------------------------------------------
  WritePtr: Takes the two integer paramters and prints them in segment
            address format, i.e. SSSS:0000
-------------------------------------------------------------------------------}

Procedure WritePtr( PtrSeg, PtrOfs : integer );
begin
  Write( HexStr(PtrSeg), ':', HexStr(PtrOfs), '   ');
end;
```

```
begin  { Main program Device_Chain }

 LowVideo;
 ClrScr;
 GotoXY(24,1);
 Write('Device Driver');
 GotoXY(2Ø,2);
 Write('Resident Header Chain');
 GotoXY(1,5);

 Writeln(' Starting      Next              Strategy   Interrupt   Device');
 Writeln(' Address     Hdr Addr   Attr   Entry Pnt   Entry Pnt   Name');
 Writeln('---------   ---------   ----   ---------   ---------   --------');

 {
   Initialize the FCB to zero and set up the NUL device driver name.  Then,
   attempt to open the device for input.  If the open is successful, the
   proper device driver pointer addresses are automatically put in the FCB
   for our use in the rest of the program.
 }

 FillChar(DeviceControlBlock,Sizeof(DeviceControlBlock),Ø);
 With DeviceControlBlock do
   begin
     Filename := 'NUL     ';
     Extension:= '   ';
     With Regs do
       begin
         AX := $ØFØØ;
         DX := Ofs(DeviceControlBlock);
         DS := Seg(DeviceControlBlock);
         MSDos(Regs);
         If (AX and $ØØFF) <> Ø
           then
             begin
               Writeln('Error in opening the NUL Device');
               Halt;
             end;
       end;
     DevicePtr     := Ptr(DevSegment,DevOffset);
     DeviceSegment := DevSegment;
     DeviceOffset  := DevOffset;
   end;

 {
   Once the proper addresses have bee established, move backward through the
   device driver chain, printing the pertinent data as we proceed.  The end of
   the device driver chain is indicated when the "next device offset" address
   is equal to -1 ($FFFF).
 }

 While DeviceOffset <> $FFFF do
   With DevicePtr^ do
     begin
 WritePtr(DeviceSegment,DeviceOffset);
 WritePtr(NextHeaderSegment,NextHeaderOffset);
 Write(HexStr(Attributes), '   ');
 WritePtr(DeviceSegment,StrategyEntPt);
 WritePtr(DeviceSegment,InterruptEntPt);

 {
   If the device is a character device (the statement below is TRUE),
   then the "device name" is a valid 8 character representation.  If
   the device is a block device (i.e., a disk drive, etc.) then this
   field usually contains a number indicating how many devices are
   supported by the driver.
 }

 if (Attributes and $8ØØØ) <> Ø
   then Write(DeviceName)
```

```
        else Write(Ord(DeviceName[1]),' Block Device Units');
      Writeln;
      DevicePtr     := Ptr(NextHeaderSegment,NextHeaderOffset);
      DeviceSegment := NextHeaderSegment;
      DeviceOffset  := NextHeaderOffset;
    end; { With DevicePtr^ }

end.  { Device_Chain }
```

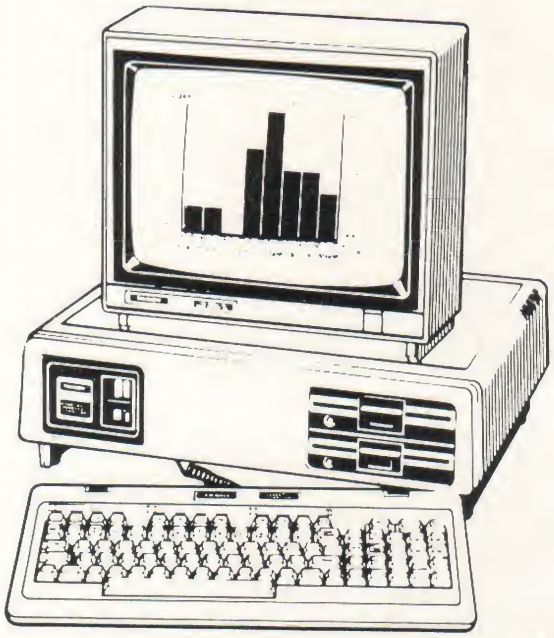
Listing 2: BASIC Program to Create PASS.SYS

```
10 CLS
20 PRINT "Creating binary file: PASS.SYS"
30 CHECK.SUM = 0
40 OPEN "R",1,"PASS.SYS",1: FIELD 1, 1 AS D$
50 READ N%
60 IF N%<0 THEN GOTO 100
70 LSET D$=CHR$(N%): PUT 1: CHECK.SUM = CHECK.SUM + N%
90 GOTO 50
100 READ CHECK2
110 IF CHECK2<>CHECK.SUM GOTO 140
120 PRINT "FILE PASS.SYS CREATED SUCCESSFULLY"
130 CLOSE: END
140 PRINT "FILE CREATION ERROR"
150 CLOSE: KILL "PASS.SYS": END
200 DATA  255, 255, 255, 255,   0, 128,  54,   0,  65,   0,  80,  65,  83,  83
210 DATA   87,  68,  36,  32, 160,   0, 129,   0, 129,   0, 129,   0, 129,   0
220 DATA  129,   0, 129,   0, 129,   0, 121,   0, 121,   0, 129,   0, 129,   0
230 DATA  129,   0, 129,   0, 129,   0, 129,   0,   0,   0,   0,   0,  46, 137
240 DATA   30,  50,   0,  46, 140,   6,  52,   0, 203, 156, 252,  86,  80,  81
250 DATA   82,  87,  85,  30,   6,  83,  46, 197,  30,  50,   0, 139,  79,  18
260 DATA  138,  71,   2,  60,  15, 119,  17,  50, 228, 208, 192, 141,  54,  18
270 DATA    0,   3, 240, 196, 127,  14,  14,  31, 255,  36, 176,   3,  41,  79
280 DATA   18, 180, 129, 235,  14, 180,   3, 235,  10,  46, 197,  30,  50,   0
290 DATA   41,  79,  18, 180,   1,  46, 197,  30,  50,   0, 137,  71,   3,  91
300 DATA    7,  31,  93,  95,  90,  89,  88,  94, 157, 203,   0,   0,   0,   0
310 DATA    0,   0,   0,   0,   0,   0, 140, 208, 139, 236, 140, 202, 142, 210
320 DATA  188,  31,   4,  80,  51, 210, 142, 194, 250,  38, 139,  30, 156,   1
330 DATA  137,  30, 223,   1,  38, 139,  30, 158,   1, 137,  30, 225,   1, 187
340 DATA  246,   0,  38, 137,  30, 156,   1, 140, 203,  38, 137,  30, 158,   1
350 DATA  251,  14,   7, 141,   6,  31,   4, 140, 202,  46, 197,  30,  50,   0
360 DATA  137,  87,  16, 137,  71,  14,  14,  31, 198,   6, 227,   1,   0, 232
370 DATA   59,   0,  23, 139, 229, 233, 139, 255, 156, 252,  86,  80,  81,  82
380 DATA   87,  85,  30,   6,  83,  14,  31, 140, 208, 139, 236, 140, 202, 142
390 DATA  210, 188,  31,   4,  14,   7,  80, 198,   6, 227,   1, 255, 232,  18
400 DATA    0,  23, 139, 229,  91,   7,  31,  93,  95,  90,  89,  88,  94,  46
410 DATA  161, 228,   1, 157, 207, 180,   9, 186,  19,   2, 205,  33, 185,  15
420 DATA    0, 176,  32, 191, 230,   1, 252, 243, 170, 232,  31,   0, 161, 228
430 DATA    1,  11, 192, 116,  16, 160, 227,   1,  60,   0, 117,  16, 180,   9
440 DATA  186, 192,   2, 205,  33, 235, 254, 180,   9, 186, 111,   3, 205,  33
450 DATA  195, 232,   4,   0, 232,  89,   0, 195,  51, 219, 180,   7, 205,  33
460 DATA   60,  13, 116,  58,  60,   8, 116,  22,  60,  27, 116,  34, 131, 251
470 DATA   15, 125, 235, 136, 135, 230,   1,  67, 180,   2, 178,  35, 205,  33
480 DATA  235, 222, 131, 251,   0, 126, 217,  75, 198, 135, 230,   1,  32, 232
490 DATA   19,   0, 235, 206, 131, 251,   0, 126, 201,  75, 198, 135, 230,   1
500 DATA   32, 232,   3,   0, 235, 240, 195, 180,   2, 178,   8, 205,  33, 180
510 DATA    2, 178,  95, 205,  33, 180,   2, 178,   8, 205,  33, 195, 160, 227
520 DATA    1,  60,   0, 117,   5, 190, 245,   1, 235,   3, 190,   4,   2, 191
530 DATA  230,   1, 185,  15,   0, 243, 166, 184,   0,   0, 116,   1,  72, 163
540 DATA  228,   1, 195,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0
550 DATA    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 112,  97, 115
560 DATA  115, 119, 111, 114, 100,  49,  32,  32,  32,  32,  32,  32, 112,  97
570 DATA  115, 115, 119, 111, 114, 100,  50,  32,  32,  32,  32,  32,  32,  13
580 DATA   10,  10,  10,  10,  10,  10,  10,  10,  10,  10,  10,  10,  10,  10
590 DATA   10,  10,  10,  10,  10,  10,  10,  10,  10,  10,  10,  80,  65,  83
600 DATA   83,  87,  79,  82,  68,  32,  86,  50,  46,  48,  13,  10,  87, 114
610 DATA  105, 116, 116, 101, 110,  32,  98, 121,  32,  74, 111, 104, 110,  32
620 DATA   72,  97, 114, 114, 101, 108, 108,  13,  10,  10,  10,  80, 108, 101
630 DATA   97, 115, 101,  32, 101, 110,  116, 101, 114,  32, 116, 104, 101,  32
640 DATA   99, 111, 114, 114, 101,  99,  116,  32, 112,  97, 115, 115, 119, 111
650 DATA  114, 100,  32,  97, 110, 100,  13,  10, 112, 114, 101, 115, 115,  32
660 DATA  116, 104, 101,  32,  82,  69,  84,  85,  82,  78,  32, 107, 101, 121
```

NOCONA

ELECTRONICS

TRS-80 Model 2000 by Tandy®



TRS-80® Model 100 by Radio Shack

MS-DOS Based

Computer System

256K MOD2000 W/MONO MONITOR	1319.00
256K MOD2000 W/COLOR MONITOR	1629.00
256K MOD 1200 HD 2 DR.	1499.00
128K MOD 1000 W/MONO 1 DR.	759.00
128K MOD 1000 W/COLOR 2 DR.	1099.00
256K MOD 1000 W/HARD DISK	1439.00
24K MOD 100 PORT	399.00
24K MOD 200 PORT	679.00
64K MOD 4D 2 DR	839.00
512K MOD 6000 2DR.	1849.00
512K MOD 6000 w/15MEG HD	2579.00
15 MEG HARD DISK	1119.00
35 MEG HARD DISK	2059.00
DMP 105 PRINTER	145.00
DMP 2200 PRINTER	1149.00
DMP 130 PRINTER	249.00
DMP 430 PRINTER	585.00
DMP 2100P PRINTER	1019.00
CGP 220 COLOR PRINTER	449.00
DWP 510 PRINTER	999.00
TRACTOR DWP 210	99.00
TRACTOR DWP II/510	159.00
TRACTOR FEED DMP2100P	99.00
PARALLEL PRINTER SWITCH	96.00
PARALLEL PRINTER CONTROLLER	179.00
PARALLEL PRINTER CABLE	29.00
DISK VIDEO INTERFACE	399.00
32K MOD 600 PORTABLE	1149.00
512K MOD 3000	1819.00

100% RS COMPONENTS NO FOREIGN DRIVES OR MEMORY FULL WARRANTY
ALL RS SOFTWARE 20% OFF CATALOG PRICE
CASHIERS CHECK OR MONEY ORDER MUST ACCOMPANY ALL ORDERS

(817) 825-4027

NOCONA ELECTRONICS • BOX 593 • NOCONA, TX 76255

```
67Ø DATA   32, 116, 111,  32,  99, 111, 11Ø, 116, 1Ø5, 11Ø, 117, 1Ø1,  58,  32
68Ø DATA   95,  95,  95,  95,  95,  95,  95,  95,  95,  95,  95,  95,  95,  95
69Ø DATA   95,   8,   8,   8,   8,   8,   8,   8,   8,   8,   8,   8,   8,   8
7ØØ DATA    8,   8,   7,  36,  13,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø
71Ø DATA   1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø
72Ø DATA   1Ø,  1Ø,   7,   7,   7,   7,   7,  89, 111, 117,  32, 1Ø4,  97, 118
73Ø DATA  1Ø1,  32, 1Ø1, 11Ø, 116, 1Ø1, 114, 1Ø1, 1ØØ,  32, 116, 1Ø4, 1Ø1,  32
74Ø DATA  119, 114, 111, 11Ø, 1Ø3,  32, 112,  97, 115, 115, 119, 111, 114, 1ØØ
75Ø DATA   32,  45,  45,  13,  1Ø,  1Ø,  89, 111, 117,  32, 1Ø9, 117, 115, 116
76Ø DATA   32, 114, 1Ø1, 115, 1Ø1, 116,  32, 116, 1Ø4, 1Ø1,  32,  99, 111, 1Ø9
77Ø DATA  112, 117, 116, 1Ø1, 114,  32,  97, 11Ø, 1ØØ,  32, 116, 114, 121,  32
78Ø DATA   97, 1Ø3,  97, 1Ø5, 11Ø,  13,  1Ø, 111, 114,  32,  99,  97, 1Ø8, 1Ø8
79Ø DATA   32, 116, 1Ø4, 1Ø1,  32, 111, 119, 11Ø, 1Ø1, 114,  32, 116, 111,  32
8ØØ DATA  112, 114, 111,  99, 1Ø1, 1Ø1, 1ØØ,  33,  13,  1Ø,  1Ø,  73,  32,  97
81Ø DATA  1Ø9,  32, 11Ø, 111, 119,  32,  68,  69,  65,  68,  33,  33,  33,  13
82Ø DATA   1Ø,  1Ø,  1Ø,  1Ø,  1Ø,   7,   7,   7,   7,   7,  36,  13,  1Ø,  1Ø
83Ø DATA   1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø
84Ø DATA   1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  84, 1Ø4,  97, 11Ø, 1Ø7
85Ø DATA   32, 121, 111, 117,  33,  13,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø,  1Ø
86Ø DATA   1Ø,  1Ø,  36,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
87Ø DATA    Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
88Ø DATA    Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
89Ø DATA    Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
9ØØ DATA    Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
91Ø DATA    Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
92Ø DATA    Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø,   Ø
93Ø DATA -1, 73765
```

Listing 3: PASS1.ASM

```
; :bk=Ø
; :ts=8
page6Ø,132

;*****************************************************************************
;
;This program shell was taken from the article "Building Device
;Drivers" by Stan Mitchell in the May 1985 issue of PC Tech
;Journal beginning on page 76.
;
;*****************************************************************************

titleDevice Driver Construction Program
nameddriver

;*****************************************************************************
;

csegsegmentpara public 'CODE'
assumeds:cseg,cs:cseg

ddriverprocfar

begin:

;
;*****************************************************************************

;
;       Initialize 8 constants for Character Device Name to ' '
;

in_namemacro
x=Ø
rept8
x=x+1
charsp%x
endm
endm

;
;Assign each letter of Character Device Name to a constant
;
```

```
 mknamemacronlist
 x=Ø
 irpcm,nlist
 x=x+1
 charequ%x,m
 endm
 endm

 ;
 ;Assign constant the ASCII value of z
 ;

 charequmacroy,z
 n&y='&z'
 endm

 ;
 ;Assign constant the ASCII value of ' '
 ;

 charspmacroy
 n&y=' '
 endm

 ;
 ;Convert ASCII constants to a string of DB constants
 ;

 ddnamemacro
 x=Ø
 rept8
 x=x+1
 chardb%x
 endm
 endm

 ;
 ;Convert an ASCII character to a DB constant
 ;

 chardbmacroy
 dbn&y
 endm

 ;
 ;Define a Device Header Structure
 ;

 devhdrmacronxtoff,nxtseg,attrib,strat,intrpt
 dwnxtoff
 dwnxtseg
 dwattrib
 dwstrat
 dwintrpt
 ddname
 endm

 ;
 ;DEF = true if this function is to be implemented
 ;NAME = name of the function to be defined
 ;ENTRY = label of the function entry point or the exit point
 ;if the function is not defined
 ;

 function macrodef,name,entry
 _&nameequdef
 if_&name
 ifndefentry
 extrnentry:near
 endif
 endif
 .&nameequentry
 endm
```

```
;
;Static request header
;

lngequØ;BYTE: first byte is length of the request header
unitequl;BYTE: unit number for block device
cmdequ2;BYTE: request command code
stateqU3;WORD: status
dos_qequ5;DWORD: DOS queue
dev_qequ9;DROWD: device queue

;
;INIT Header
;

unitsequl3;BYTE: number of units
brkoffequl4;DWORD: ending address for resident code
brksegequl6
bpbequl8;DWORD: pointer to the BIOS Parameter Block
paramsequl8;DWORD: pointer to the string following the
;   "device=' command in CONFIG.SYS

;
;Non-destructive read -- no wait
;

prechareqUl3;BYTE: one look-ahead character

;
;Input, Output, IOCTL In, IOCTL Out
;

mediaequl3;BYTE:  media descriptor byte
bufequl4;DWORD: pointer to buffer address
cntequl8;WORD:  byte/sector count
startequ2Ø;WORD:  starting sector number

;
;Status word byte definitions  -- most significant byte
;

reqdonequØlh;Requested command complete
busyequØ2h;Device busy
errorequ8Øh;Device error encountered

;
;Status word error codes -- least significant byte
;

notrdyequØ2h;Device is not ready
unkcmdequØ3h;Unknown device command
nopaperequØ9h;Printer out of paper
wrfltequØah;Device write fault
rdfltequØbh;Device read fault
failequØch;General device failure

;*****************************************************************************
;
;DEVICE DEFINITION FILE
;
;*****************************************************************************

trueequl
falseequØ

in_name;Initialize device name to blanks

;*****************************************************************************
;
;Select DOS version number:
;For DOS 2.x set DOS2_x to true
;For DOS 3.x set DOS2_x to false
```

```
DOS2_xequfalse
DOS3_Øequtrue


;*****************************************************************************
;
;Enter the device driver name here in place of the X's
;(1 to 8 characters in uppercase)
;
;mknameXXXXXXXX

mknamePASSWD$


;*****************************************************************************
;
;Select the attributes for the device from the following table:
;
;Defined attribute bits:

chrequ8ØØØH;Character Device
ioctlequ4ØØØH;I/O Control Functions supported
ibmequ2ØØØH;Block Device is IBM format

ifDOS3_Ø;Open, Close, and Removable Media
ocremequØ8ØØH;  is supported for DOS 3.x
endif

specialequØØ1ØH;Special Device using INT 29H handler

clockequØØØ8H;Device is current CLOCK device
nulequØØØ4H;Device is current NUL device
stdoutequØØØ2H;Device is current standard output device
stdinequØØØ1H;Device is current standard input device

;Example:  chr+special+stdout+stdin
;defines a character device using a special INT 29H handler
;to support STDIN and STDOUT and this will generate an
;attribute word of 8Ø13H
;
;*****************************************************************************
;
;Edit the line below to select the correct attributes for your device
attribequchr;Device is a character device

;*****************************************************************************
;
;Each line in the function table contains 3 arguments:
;
;#1 is true if the function is defined, false if not
;#2 is the name of a valid function to which the true or false
;   value applies
;#3 is the label which is defined as the entry point for this function.
;   If the function is not to be implemented, "done", "done2", "xfer",
;   or "invalid" should be used.  Also note that the Function macro
;   defines the entry point label as "external" if it is undefined in
;   the current program.

function true,init,passinit
function false,media_check,done
function false,build_bpb,done
function false,ioctl_in,done
function false,input,done
function false,nd_input,done
function false,in_stat,done
function false,flushin,done
function false,output,xfer
function false,voutput,xfer
function false,out_stat,done
function false,flushout,done
function false,ioctl_out,done
ifDOS3_Ø
function false,open,done
```

It's the Holmes Engineering/PCSG "chipmunk"

PORTABLE Disk Drive

AS FAST AS RAM

for the Model 100/200

NEW LOW PRICE
$499.

Includes a library of six powerful programs

NOW AVAILABLE ON THE M-200

ASK ABOUT BUDGET MODEL $399.

Uses the main menu concept. You see the disk directory instantly, arranged on your M-100 screen like your main menu. Just move the widebar cursor and transfer files with a function key. You can run a file directly from the diskette with the ENTER key. Uses 3½" microfloppy diskettes that have a rigid plastic casing and a metal core. They're tough and nonflexible. You can carry several in a shirt pocket without damage. There's 358K on a diskette. Ten of these in your briefcase and you've got 3½ megabytes.

Drive weighs only three lbs. and it works directly from the 110 outlet and recharges at the same time. It recharges in six hours with thousands of pages transferred between charges. It's compact, with dimensions of 2¼" x 5½" x 7.5"; and fits easily into your briefcase along with your Model 100 or 200.

Machine code programs, BASIC programs, *Lucid* files and documents all are saved and retrieved with no protocol–instantly, ready to run.

In a special association, Holmes Engineering and PCSG have worked together combining the hardware knowledge of Holmes and the software expertise of PCSG. The result is a product that can only be regarded as excellent.

You see the disk directory instantly; works just like the main menu

Here is what is really exciting. The portable disk drive has Random Access. Included as part of the operating system in the drive (ROM) is a very powerful disk BASIC.

This means that you can have BASIC programs that will access the diskette and read and write records directly on the diskette.

Just imagine yourself with this kind of capability.

Database–The portable disk drive stores your mailing list, inventory items, part #s and descriptions or any other data that you need to recall.

358K on a diskette

Invoice (purchase order)–At the touch of a button you can print out your sequentially numbered, professionally done invoices. This is truly professional invoicing capability.

Purchase orders are just as easy.

Sort – This excellent utility allows you rapid sorting of any records you have compiled. You can write the newly sorted list back in the same file on the diskette or to a new file.

Telecom interface–If you are a user that likes to access other computers or databases (for example CompuServe) by telephone then this powerful facility alone is worth the price of the disk drive. You can automatically download and upload information onto the diskette.

Calendar – Everyone who has seen this program has said, "This is the first calendar/diary/scheduler on any computer anywhere that I can use. It is so functional."

The calendar program is usable for two reasons, first it is designed correctly, and second you have the memory (358K) on the diskette to log and access a tremendous amount of notes over a long period of time.

Personal Finance Manager – This wonderful program truly lets you keep track of your finances.

All your records are kept on the diskette. Bank accounts (checking and savings) and charge accounts such as MasterCard and Visa.

We at PCSG believe we have the ultimate Model 100 system, the Portable Disk Drive plus the *Lucid* spreadsheet on snap-in ROM, *Write ROM* word processing and the new 64K RAM expansion now available from PCSG.

We want you to find out for yourself at no risk. If you aren't totally satisfied within 30 days, simply return the disk drive for a full refund. Priced at $499.00, including the software library. MasterCard, Visa, COD.

PORTABLE COMPUTER SUPPORT GROUP

1-214-351-0564

11035 Harry Hines Blvd., #207
Dallas, Texas 75229

PCSG offers a 30-day/money-back trial!

```
function false,close,done
function false,removable,done
endif

;
;*****************************************************************************

devhdr-1,-1,attrib,strat_ent,int_ent

;
;Function Table Definition
;

funtbllabelbyte;CodeFunction
; ************************************
dw.init;  ØInitialization
dw.media_check;  1Block device media check
dw.build_bpb;  2Block device build BPB
dw.ioctl_in;  3IOCTL Input
dw.input;  4Device input command
dw.nd_input;  5Non-destructive input
dw.in_stat;  6Device input status
dw.flushin;  7Flush input
dw.output;  8Output to device
dw.voutput;  9Output to device with verify
dw.out_stat; 1ØOutput status
dw.flushout; 11Flush output
dw.ioctl_out; 12IOCTL output

ifDOS3_Ø
hifuncequl5
dw.open; 13Open Device
dw.close; 14Close Device
dw.removable; 15Removable media check (Block Device)

else

hifuncequl2

endif


publicreqhdr,done,invalid,done2,errout,xfer
;
;Define the request header pointer
;

reqhdrlabeldword
reqhdrodw?;header offset value
reqhdrsdw?;header segment value

;
;Define the device Strategy Entry Point
;

strat_ent:
movcs:reqhdro,bx;Save the offset of the request header ptr
movcs:reqhdrs,es;Save the segment of the request header ptr
ret;Far return

;
;Define the device Interrput Entry Point
;
;Note: DOS's stack allows saving the registers but if your routine
;needs more stack space than that, you should define a local stack
;

int_ent:
pushf;Save the flag status
cld;Set UP direction for string operations
pushsi;Save processor registers
pushax
pushcx
```

```
pushdx
pushdi
pushbp
pushds
pushes
pushbx

ldsbx,cs:reqhdr;DS:BX points to the request header
movcx,[bx+cnt];Get the byte count for the operation
moval,[bx+cmd];Get the device command byte
cmpal,hifunc;Test for an invalid code
jainvalid;Post error and return if out of range
xorah,ah;Double the command code for an offset
rolal,1;   into the jump table
leasi,funtbl;Set up pointer to function table
addsi,ax;Get correct entry
lesdi,dword ptr [bx+buf];ES:DI points to command buffer addr
pushcs
popds;Set DS register properly
jmpword ptr [si];Enter proper command routine from table

invalid:
moval,unkcmd;Set unknown command error
sub[bx+cnt],cx;Zero the byte count

errout:
movah,error+reqdon;Set error and done bits
jmpshort exit

done2:
movah,reqdon+busy;Set done and busy bits
jmpshort exit

xfer:
ldsbx,cs:reqhdr;Get pointer in DS:BX for request header
sub[bx+cnt],cx;Return number of bytes transferred

done:
movah,reqdon;Set done status bit

exit:
ldsbx,cs:reqhdr;DS:BX set to request header pointer
mov[bx+stat],ax;Set status return value

popbx;Restore processor registers
popes
popds
popbp
popdi
popdx
popcx
popax
popsi
popf
ret;Return to operating system

;
;*************************************************************************

ddriverendp

csegends

endbegin
```

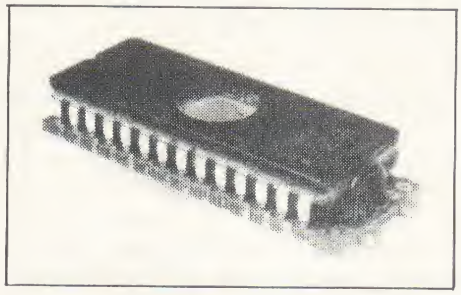
Listing 4: PASS2.ASM

```
; :bk=Ø
; :ts=8
```

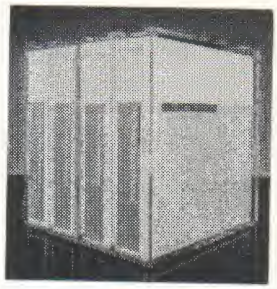
```
page60,132
namepassword
titlePassword Validation Routine

.sall

;*****************************************************************************
;
;PASSWORD --
;
;Read a password from the keyboard and return control to the
;user if the password is valid.  If the password is not valid,
;"hang" the system and force a system reset.
;
;PASSWORD uses only DOS function calls less that 0CH to allow
;it to function as an "installed" device driver and prevent
;user interruption of the validation routine during boot-up
;of the system.
;
;Using these MS-DOS normal function calls, PASSWORD should
;operate on any machine using the MS-DOS operating system.
;This means that you should be able to use this routine on
;all non-compatible machines such as the WANG and ZENITH
;systems.
;
;
;
;Written by:
;
;LCDR John B. Harrell, III
;1519-A Carswell Circle
;Bolling Air Force Base
;Washington, DC   20336
;
;
;Date written:
;22 January 1986
;
;Modifications:
;7 February 1986 -- added additional code to clean
;up the password handling and make this routine
;compatible with all DOS releases.  PC-DOS
;version 2 did not like the special characters
;in the device driver header.
;
;9 February 1986 -- added a second password routine
;which is accessible after booting the system.
;Any user program can require password valid-
;ation to continue by using this Interrupt 67H
;function call.  This call will not force a
;system reset.
;
;*****************************************************************************
;*****************************************************************************
;
;Constant Definitions
;
;*****************************************************************************

;
;Static request header
;

lngequ0;BYTE: first byte is length of the request header
unitequ1;BYTE: unit number for block device
cmdequ2;BYTE: request command code
statequ3;WORD: status
dos_qequ5;DWORD: DOS queue
dev_qequ9;DROWD: device queue

;
;INIT Header
;
```

We would like to take this opportunity to introduce ourselves. Although **Ross Computer Services** is a new name to the pages of this magazine, we are *NOT* new to computers. We have been suppling computer dealers with many of the products they sell for several years. We have decided to end that relationship, and to sell directly to you, the end user.

ROMDSK. 8K or 32K CUSTOM expansion rom for the TRS-80 Model 100. Custom programmed with YOUR programs.
8K . . . . . **$49.95** 32K **$69.95**
Call or Write **ROSS COMPUTER SERVICES**
For Ordering Information



INNOVATIVE™ Software



The SMART Software System, from Innovative Softwear.

The SMART DATA MANAGER	**$495.00**
The SMART WORD PROCESSOR	**$295.00**
The SMART SPREADSHEET (With Graphics)	**$395.00**
The SMART SPELLCHECKER	**$139.00**
The SMART SYSTEM*	**$895.00**

*The SMART Software System includes: The Smart Data Manager, The Smart Spreadsheet with Graphics, and The Smart Word Processor, and can only be sold as a system, not as individual products. The smart Software System includes built in asynchronous communications and personal time management programs.

BORLAND INTERNATIONAL

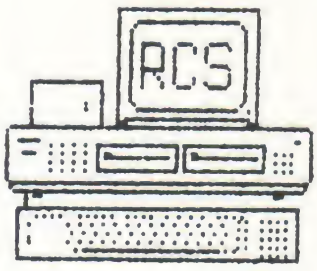
Turbo Tutor	**$ 34.95**	Superkey	**$ 69.96**
Turbo Pascal 3.0	**$ 69.95**	SideKick (copy-protected)	**$ 54.95**
Turbo Pascal with 8087	**$109.90**	SideKick (NOT copy-protected)	**$ 84.95**
Turbo Pascal with BCD	**$109.90**	Traveling SideKick (when available)	**$ 69.95***
Turbo Pascal 3.0 with both 8087 & BDC	**$124.95**	Traveling SideKick with SideKick (NCP)	**$125.00**
Turbo Database Toolbox	**$ 54.95**	Turbo Lightning	**$ 99.95**
Turbo Graphix Toolbox	**$ 54.95**	Reflex: The Analyst	**$ 99.95**
Turbo Editor Toolbox	**$ 69.95**		
Turbo Gameworks	**$ 69.95**		

*Traveling SideKick should be available by the time you read this ad.

MERCURY 300/ 1200 MODEM

Mercury 300/1200 auto-answer/auto-dial, HAYES compatible MODEM. Uses the industry standard AT command set.
SPECIAL **$199.95**

Ross Computer Services is a division of a company that has been in business for over 54 years. We are not likely to fold our tents, and steal into the night. We will sell only the *highest quality hardware and software.* Though our initial offering is limited, we will be constantly adding to our product line, again, adding only products of the highest quality.



ROSS COMPUTER SERVICES

(816) 363-2545

VISA® MasterCard®

Hours: 8:30 to 5:00
After 5:00 you will reach our order line BBS

1800 W. 91st Place • Kansas City, MO 64114

```
unitsequl3;BYTE: number of units
brkoffequl4;DWORD: ending address for resident code
brksegequl6
bpbequl8;DWORD: pointer to the BIOS Parameter Block
paramsequl8;DWORD: pointer to the string following the
;   "device=' command in CONFIG.SYS

;
;Non-destructive read -- no wait
;

precharequl3;BYTE: one look-ahead character

;
;Input, Output, IOCTL In, IOCTL Out
;

mediaequl3;BYTE:  media descriptor byte
bufequl4;DWORD: pointer to buffer address
cntequl8;WORD:  byte/sector count
startequ2Ø;WORD:  starting sector number

;
;Status word bite definitions  -- most significant byte
;

reqdonequØ1h;Requested command complete
busyequØ2h;Device busy
errorequ8Øh;Device error encountered

;
;Status word error codes -- least significant byte
;

notrdyequØ2h;Device is not ready
unkcmdequØ3h;Unknown device command
nopaperequØ9h;Printer out of paper
wrfltequØah;Device write fault
rdfltequØbh;Device read fault
failequØch;General device failure


lfequlØ;Line feed code
crequl3;Carriage return code
bsequ8;Backspace code
bellequ7;Bell tone code (beep)
ulineequ'_';Underline character
voidlineequ27;Character used to void the line
blankequ' ';Blank character
echo_charequ'#';Password echo character
eosequ'$';DOS End-of-string character
pass_lengthequl5;Maximum password character length
dosintequ21h;MS-DOS function cmd processor inrpt
vector67equ67h*4;Offset in page Ø for interrupt 67


;*****************************************************************************
;
;Macro Definitions
;
;*****************************************************************************

;charin
;Reads the next character available from the keyboard and
;returns it in the AL register.  The character is not echoed
;to the display and no checking for control characters is
;performed.

charinmacro
movah,7
intdosint
endm
```

```
;dispstr
;Display the string located at the data offset passed as a
;parameter to the macro.  The string is displayed until a
;terminal '$' character is located (standard MS-DOS terminal)

dispstrmacrostr
movah,9
movdx,offset str
intdosint
endm

;dispch
;Display the character passed as a parameter on the video
;display

dispchmacrochr
movah,2
movdl,chr
intdosint
endm


;*****************************************************************************
;
;Main Code Segment
;
;*****************************************************************************


csegsegmentpara public 'CODE'
assumeds:cseg,cs:cseg,es:cseg,ss:cseg

extrnreqhdr:dword
extrndone:near
publicpassinit


;*****************************************************************************
;
;passinit --
;Entry point for user password entry during system
;startup.    If the password is not valid, the system
;will hang up and require a reset
;
;*****************************************************************************

passinit procfar

movax,ss;Set up a temporary stack and save SS
movbp,sp;Current SP to BP - BP is not used
movdx,cs
movss,dx
movsp,offset stack;Set new stack offset in this Code segment
pushax;Save stack segment on new stack

xordx,dx;Clear DX register
moves,dx;Point to first page segment
cli;Disable interrupts momentarily
movbx,es:[vector67];Get the offset of current intr
movword ptr [oldint67],bx;Save address
movbx,es:[vector67+2]
movword ptr [oldint67+2],bx; and segment for later use

movbx,offset passint;Set new interrupt offset
moves:[vector67],bx
movbx,cs;Set proper segment address
moves:[vector67+2],bx
sti;Re-enable interrupts

pushcs
popes;Set ES register to Code Segment
```

```
leaax,cs:word ptr rel_addr;Terminal address for device driver
movdx,cs;Segment of terminal byte
ldsbx,cs:reqhdr;DS:BX points to INIT request header
mov[bx+brkseg],dx;Store segment address for return
mov[bx+brkoff],ax;Store offset
pushcs
popds;Restore DS register

movbyte ptr[pwflag],Ø;Set up scan with boot password
callpassword;Read actual password from user

popss;Restore original stack segment address
movsp,bp;Restore stack offset pointer
jmpdone;Return to device driver

passinit endp


;*****************************************************************************
;
;passint --
;This is the interrupt handler entry point for the newly
;defined INT 67H call (see above).  First, call the old
;vector and then handle the password.  A valid password
;will return Ø in the AX register, nonvalid will return
;-1 in AX.  All other registers will remain unchanged.
;
;*****************************************************************************

passintprocfar

pushf;Save the flag status
cld;Set UP direction for string operations
pushsi;Save processor registers
pushax
pushcx
pushdx
pushdi
pushbp
pushds
pushes
pushbx

pushcs;Set up DS segment
popds

movax,ss;Set up a temporary stack and save SS
movbp,sp;Current SP to BP - BP is not used
movdx,cs
movss,dx
movsp,offset stack;Set new stack offset in this Code segment
pushcs
popes;Set ES and SS registers to Code Segment
pushax;Save stack segment on new stack

movbyte ptr[pwflag],-1;Set call for user password
callpassword;Read actual password from user

popss;Restore original stack segment address
movsp,bp;Restore stack offset pointer

popbx;Restore processor registers
popes
popds
popbp
popdi
popdx
popcx
popax
popsi

movax,cs:word ptr[retflag];Get return result
```

```
    popf
    iret

    passintendp


    ;*************************************************************************
    ;
    ;password
    ;Read password from user for initialization and subsequent
    ;calls for file input
    ;
    ;*************************************************************************

    password procnear

    dispstrsignon;Display sign-on message

    movcx,pass_length;Clear the input string to blanks
    moval,blank
    movdi,offset pword;ES:DI points to target area
    cld;Set string move direction = UP
    rep stosb;Clear the string

    callget_password;Get and validate the password from user

    movax,word ptr[retflag];Get return code
    crax,ax;Check return code for good password
    jzpw_ok;Password validates -- release system for user

    moval,byte ptr[pwflag];Test for initial or interrupt password
    cmpal,Ø;If not equal, Interrupt 67H entry
    jnepw_exit;Set return code and exit

    dispstroops_msg;Let the user know he is FUBAR
    pw_bad:jmpshort pw_bad;Initial entry - hang up the system in a loop

    pw_ok:dispstrthanks_msg;Thank user and sign-off

    pw_exit:
    ret

    password endp


    ;*************************************************************************
    ;
    ;get_password
    ;Get the password from the user and then validate it
    ;
    ;*************************************************************************

    get_password proc near

    callread_pw
    callvalidate
    ret

    get_password endp
    ;*************************************************************************
    ;
    ;read_pw
    ;Read the password from the user and store the characters in
    ;the array located at offset "pword".
    ;
    ;Read_pw allows use of the backspace key to remove one character
    ;at a time from the input line or use of the Escape key to void
    ;the entire input and return to the start of the entry.
    ;
    ;This routine will exit only after the Enter (Return) key has
    ;been pressed.
    ;
    ;*************************************************************************
```

```
read_pwprocnear

xorbx,bx;Clear the array index

read_pwl:
charin;Get the next character from the user
cmpal,cr;Test for carriage return
jzshort read_pwx;Exit routine if found

cmpal,bs;Test for backspace key
jzshort read_bs;Delete one character
cmpal,voidline;Test for line delete
jzshort read_void;Delete entire line

cmpbx,pass_length;Check for at end of allowable entry
jgeread_pwl;Yep -- loop for carriage return
movbyte ptr pword[bx],al;Store character
incbx;Bump pointer
dispchecho_char;Display echo character for user response
jmpread_pwl;Continue to read characters

read_bs:
cmpbx,0;Check for at start of line
jleshort read_pwl;Yep -- no backspace allowed
decbx
movbyte ptr pword[bx],blank;Zap this character
callerase;Update screen
jmpread_pwl;Back for more characters

read_void:
cmpbx,0;Check for at start of line
jleshort read_pwl;Yep -- no backspace allowed
decbx
movbyte ptr pword[bx],blank;Zap this character
callerase;Update screen
jmpread_void;Erase remaining characters

read_pwx:
ret

read_pwendp
;*****************************************************************************
;
;erase
;Erase the last echo character from the video screen by
;sending the sequence of characters: backspace, underline,
;followed by backspace.
;
;*****************************************************************************

eraseprocnear

dispchbs;Backspace one character
dispchuline;Send one underline character
dispchbs;Backspace one character
ret

eraseendp


;*****************************************************************************
;
;validate
;Validate the password entered by the user, comparing it
;byte for byte with the constant stored at offset "pconst"
;the string must be an exact match to validate, case considered
;
;If the password is valid, set "retflag" zero, otherwise
;non-zero.
;
;*****************************************************************************
```

```
validate proc near

mov al,byte ptr[pwflag];Test type of entry - initial or intr
cmp al,0;If not equal, then intr entry
jne short val_intr
mov si,offset initpw;set DS:SI to proper constant password
jmp short val_0;Check password

val_intr:
mov si,offset userpw

val_0:
mov di,offset pword;ES:DI points to user input
mov cx,pass_length;Get compare length
repz cmpsb;Compare strings
mov ax,0;Initialize return flag
jz short validate_x;Compare was ok -- password checks
dec ax;Set return flag

validate_x:
mov word ptr[retflag],ax;Set return code
ret

validate endp
;*****************************************************************************
;
;Data Definition Area
;
;*****************************************************************************

oldint67 dd?

pwflagdb0;Flag for which type of password
retflagdw0;Return result word for INT 67H

pworddbpass_length dup(?);User input area

initpwdb'dos wizz';Constant password value
iplenequ$-initpw
db(pass_length-iplen) dup(blank);fill remaining length

userpwdb'jbhiii';Constant password value
uplenequ$-userpw
db(pass_length-uplen) dup(blank);fill remaining length

signondbcr,25 dup(lf)
db'PASSWORD V2.0',cr,lf
db'Written by John Harrell',cr,lf,lf,lf
db'Please enter the correct password and',cr,lf
db'press the RETURN key to continue: '
dbpass_length dup(uline)
dbpass_length dup(bs)
dbbell,eos

oops_msg dbcr,25 dup(lf),5 dup(bell)
db'You have entered the wrong password --',cr,lf,lf
db'You must reset the computer and try again',cr,lf
db'or call the owner to proceed!',cr,lf,lf
db'I am now DEAD!!!',cr,5 dup(lf)
db5 dup(bell),eos

thanks_msg dbcr,25 dup(lf)
db'Thank you!'
dbcr,10 dup(lf),eos

dw64 dup(?)
stackequ$

rel_addr = $;Device driver can release code after this

csegends

endpassword
```

PCM

This month, we introduce you to a series on beginning BASIC programming . . .

Welcome to BASIC

By Richard A. White

An important question these days is what is BASIC's place in the world of microcomputing? Over the past few years the IBM PC and its compatibles have arrived, followed by a vast outpouring of software of every description. Each machine comes with a BASIC language interpreter on its DOS disk.

It is likely that only a small percentage of these interpreters have been loaded for the purpose of doing programming. A much larger percentage have been loaded for the purpose of running some existing BASIC program. This only reflects that early microcomputer owners were as interested in the machines themselves as they were in running applications, while more recent purchasers are buying machines to run applications only.

The market shift is also exemplified by the documentation supplied with the machine. When I got my first microcomputer in 1980, its manual was mainly concerned with how to program in BASIC. The Tandy 1000 comes with *DeskMate* and its documentation, enough imformation on MS-DOS for the user to get *DeskMate* up and running, and virtually no information on the BASICA interpreter that is on the DOS disk. Most of the thousands of folks who purchased Tandy 1000s before Christmas 1985 are probably unaware that they have BASICA on their DOS disk, or even what BASICA is for.

The role of BASIC has obviously diminished, but it is by no means a dead language. It comes with all Tandy machines and is available for developing those special applications for which software cannot be purchased. It is much easier to use for small programs to accomplish specialized tasks that cannot be done in a spreadsheet, for example, than most any other language.

Thumb through a few PCMs and you'll see most of the programs are written in BASIC, with good reason. It is the language available to everyone. A published BASIC program is potentially usable by any PCM reader, while a program in any other language is usable by only a small percentage of the readership.

Given this situation, we feel it is useful to provide a series of columns on "Beginning BASIC Programming." Early in PCM's history, I did a similar series on Model 100 BASIC. This time we will use BASICA, which comes with the Tandy 1000, 1200, 2000 and 3000. I will focus on a simple subset of the commands available and avoid detailed description of seldom-needed options.

Over the years, BASIC has been criticized by computer scientists and supporters of competing languages. Some of this criticism was justified. But BASICA for the Tandy MS-DOS machines is the result of years of refinement and is now much more powerful than any BASIC available, even five years ago, on any machine. Still, it is an interpreted language. This means BASICA must take each command it comes to in the program and look up a machine language routine to perform that command; this takes time. Borland's Turbo PASCAL, which has sold over 400,000 copies, is BASICA's major competitor. Its final product is machine language, where the machine language routines to perform each command are included directly in the program to be executed. This is called a *compiled program* and, since the computer is not continually looking things up, it is lightning fast.

However, many computer tasks do not need that speed. A data entry program is human paced, for example. And, the way the program is written can have a drastic effect on its speed. I am reminded of a friend who came to me with a program to analyze survey data. It was very slow under BASIC and he wanted a machine language version to speed things up. As I considered how to do this, it occurred to me that I could use arrays in BASIC to model how a machine language program might work. It turned out that the rewritten BASIC program was more than 10 times faster than the first version, and I never had to write a machine language version.

This paid off later. The original program was on a non-MS-DOS machine. We came to an application that was to run on an IBM PC. With minor editing, we were able to bring the BASIC programs over to the PC. If there had been a machine language module it would have needed total rewriting.

The single most valid criticism of BASIC is that it imposes no structure or discipline on the programmer. A program can be written almost any way you want. The simple can be made complex if you do not do some planning first and impose some self-discipline. PASCAL was written partly in reaction to BASIC's total lack of inherent structure. PASCAL demands specific structuring for the program to work. All variables must be defined at the beginning of the source code or at the beginning of the procedure or function where they are used. All procedures and functions that the main procedure uses must be in front of the main procedure in the source code, or there must be instructions on how PASCAL finds the procedure. How the procedure itself must be constructed is also defined.

BASIC requires none of this. On the other hand, the more structure the

---

Richard White has a long background with microcomputers and specializes in BASIC programming. He has authored numerous programs and articles. His work has also appeared in PCM's sister publication, THE RAINBOW.

programmer can bring to a BASIC program, the better. One way to do this is to develop standard subroutines and put them in standard locations in each program. For example, there are many times when it is handy to get a single character from the keyboard, which may be the starting letter of a menu choice. The program then branches according to what the letter is. The computer may be in lowercase or uppercase. It simplifies the code if it needs to look for only a lowercase or uppercase character, but not both.

The subroutine listed here loops until it gets a character. It checks if the character is lowercase, ASC(I$)>96. If so, 32 is subtracted from the ASCII value of the character and the resulting number is converted back to character form with I$=CHR$(ASC(I$)-32). The program returns to the line that called the subrouting. If the character is not lowercase, no conversion is made and the return is made immediately.

```
1 I$=INKEY$:IF I$="" THEN 1
ELSE:IF ASC(I$)>96 THEN
I$=CHR$(ASC(I$)-32):RETURN
ELSE RETURN
```

In this bit of program, or code, the computer gets a user input from the keyboard, tests it, and causes a conversion to be made if the test proves true. True or false, control is returned to the line called the subroutine. Not bad for one line of code. It even makes sense when divided into little pieces. Perhaps the secret to programming is to "divide and conquer."

Perhaps? Nay, it's a certainty. If you think of the programs as functional little pieces, you will do a lot better. PASCAL was also written for use in teaching programmers to structure their programs. In fact, you cannot write in PASCAL without adhering to strict structural rules. In BASIC you have a choice of writting free-spirited spaghetti-like programs or relatively structured ones. Reading a "spaghetti" program is like trying to read an upside-down road map in the light of a new moon at midnight. New programmers seldom know where they are going with a program and halfway through may get bogged down figuring where they've already been as well. Such is part of the learning process, so don't lose heart.

Program structure means program organization. Most languages require that variables, files, data structures and other attributes be declared at the beginning. Generous use of remarks is encouraged and specific indentation formats are strongly suggested. Structuring also deals with what should be in subroutines, what should be in the main procedure and how the procedures flow. Much of this is optional in BASIC; some is just good practice in any language.

Clarity should be as important a goal in BASIC as it is in other languages. There are a number of ways to write clear programs. I use the following procedures and find them valuable.

1) Define specific program functions and put the code for each function in its own module with introductory REMs. Assign a specific set of lines to a module. Blocks of 100 lines are convenient and will meet most needs. You will always know a module begins at an even hundred and can go right to the one you want.

2) Minimize looping back. The procedure should flow from start to end and loop only to repeat the routine or a portion of it.

3) Use IF . . . THEN . . . ELSE to minimize jumping forward. Many times all the code for the choices can be contained after THEN and ELSE on one line so the procedure can continue on the next line. The INKEY$ example clearly demonstrates this. Some BASICs don't have ELSE. Apple doesn't and Apple owners pay extra for its lack. To illustrate, which is clearer to you?

```
10 IF X=0 THEN PRINT "FALSE":X
=1 ELSE PRINT "TRUE"
20 END
```

or

```
10 IF X=0 THEN 30
20 PRINT "TRUE:" GOTO40
30 PRINT "FALSE" :X=1
40 END
```

4) Put all subroutines in one of two places. Frequently called subroutines, which include those that affect program speed, should be at the front of the program. I reserve lines 1 to 99 for these. These should not be more than three or four lines long. Line-number spacing of two is good. Putting these here serves two functions. First, the computer finds them quickly when speed counts. You also save bytes since the number in the GOSUB is only one or two bytes long.

Infrequently called routines, particularly program initialization code, should be at the end of the program. Each time a subroutine is called or the computer is sent to a specific line, it starts at the beginning of the program and searches until it finds that line. The fewer lines it passes to find the needed line, the faster the search will be. It follows that speed is compromised if the computer is continually searching over code it has used and will not use again. Clarity comes from having only two places to look for subroutines.

The one exception I make is to place a major subroutine at the end of the module that calls it when only that module uses it.

5) The same reasoning just explained also applies to the ordering of main program modules, provided they are called separately. Those used most frequently are put in front of the occasionally used ones. In a file program, the input module is used far more than the save-to-disk module and should come to the front. Modules that are used in order should be placed in order in the program.

6) Menus should be placed where they are used in the program. A menu's text provides valuable information on the branching of the program that follows. If menu choices are numbered from one up in sequence, the the ON I GOTO *xxx,yyy,zzz* or ON I GOSUB *xxx,yyy,zzz* commands can be used. It is easy to read a listing, see which number corresponds to the code block you want, drop down to the ON I . . . statement and count across to find the target line number.

Though I have discussed program structuring for purposes of easy trouble-shooting and modification, the structure proposed has program speed firmly in mind as well. Memory use is another consideration. Clarity, memory and speed are like three corners of a triangle — you cannot be at all three points at once. However, choices can be made that shorten the sides of the triangle to get you closer to where you want to be. It takes careful thought and planning at the start so you don't end up redoing too much.

The advent of cheap memory has distorted the triangle in recent years. There is no longer such a premium placed on memory conservation. Unfortunately, the relaxation of this discipline has led to sloppy programming and large programs that are slow and sometimes buggy. Small is still good.

Don't be upset when you don't get what you want the first or second time. Remember, all good commercial programs have version numbers and Version 1.0 is the first one offered for sale, not the first one of the development process. PCM

Sneak this program into your friends' Model 100s and see their reaction when their computer apparently loses its memory

April Fool!

By Art Chatham

An imp is defined as a "mischievious child or a young demon." This BASIC program is an imp. It is small (occupying only 336 bytes when the unnecessary spaces are removed) but it can render a near heart-stopping fear. When loaded in a portable computer, *April* invisibly waits for the proper time to strike, then all RAM appears to be erased! A few seconds after the victim desperately presses any key, he is reminded of the date and all is restored.

You will need no more than five minutes to place and trigger this April Fool's prank. Load and store APRIL.BA from a cassette, then load and run *Bury*. Kill BURY.BA, remove the cassette and wait for the reaction on the first day of April.

April becomes the IPL, the program that is invoked on a warm startup. It immediately calls the menu routine, until the date becomes April 1. Then, the pseudo menu is displayed, which indicates all user files and programs are gone! This display remains constant (except for the normal clock) until some seconds after a key is pressed. The screen is then blacked out for another time delay before printing APRIL FOOL. Finally, the real menu is called and the victim starts breathing again.

The *April* program is basically a series of appropriate PRINT@ statements with a few commands that might not be readily understood. In Line 2, POKE 63056,128 disables the BREAK command (later in Line 8, POKE 63056,0 reenables). CALL 16959 locks the screen from scrolling when printing to the bottom line. CALL 17001 causes the following print characters to be reverse printed (white on black) and CALL 17006 returns to normal print. In Line 5, 29638 Bytes free is correct for a 32K portable computer. For a 24K, change to 21446 Bytes free; a 16K, 13254 and 8K, 5062. CALL 23920 causes the expected clock-calendar to print and update until a key is pressed.

If your intended victim already uses an IPL program, change Line 1 from THEN MENU to THEN RUN "ipl.BA", where ipl is the existing IPL program name. If you're not sure, the computer stores the IPL program name at addresses 64175 to 64183, — PEEK at them.

BURY gets a little more interesting. Lines 1 to 3 search the user part of the directory for *April*. If *April* isn't found (load and save it first), you're alerted that all is not well. When *April* is located, Line 5 sets Bit 3 in the directory flag to make the filename invisible to menu. Line 6 establishes *April* as the IPL program and announces that *Bury*'s skullduggery is completed. □

Art Chatham is an electronics engineer for the Naval Ship Weapons Systems Engineering Station in Port Hueneme, California. His hobbies include camping, leather carving and woodworking.

PCM BAR CODED LISTING

Listing 1:

```
1 IFDATE$<>"Ø4/Ø1/85"THENMENU
2 POKE63Ø56,128:CLS:BEEP:CALL16959:PRINT
@27,"(C) Microsoft"
3 CALL17ØØ1:PRINT@4Ø," BASIC     ":CALL17
ØØ6:PRINT@51,"TEXT      TELCOM     ADDRSS
    SCHEDL"
4 FORX=91TO271STEP1Ø:PRINT@X,"-.-":NEXT
5 PRINT@28Ø,"Select: _":PRINT@3Ø2,"29638
 Bytes free"
6 CALL23920:GOSUB9
7 LINE(Ø,Ø)-(239,63),1,BF:GOSUB9
8 PRINT@135," APRIL FOOL ":BEEP:GOSUB9:P
OKE63Ø56,Ø:MENU
9 FORX=1TO2ØØ:X=X^3/X^2:NEXT:RETURN
```

PCM BAR CODED LISTING

Listing 2:

```
1 FOR X=6393ØTO64139STEP11
2 A$="":FOR Y=3TO7:A$=A$+CHR$(PEEK(X+Y))
:NEXT
3 IF A$="APRIL"THEN W=X:GOTO5
4 NEXT:PRINT "ERROR-Cannot Find APRIL in
 Directory":END
5 N%=PEEK(W)OR8:POKEW,N%
6 IPL"APRIL.BA":PRINT"FINISHED--":PRINT"
now KILL  BURY.BA"
```

PCM

LeScript — A Hard-to-Beat Word Processor

Software · · **1000/1200/3000**

Word processing is one of the most important applications for microcomputers, and because of the dozens of good programs available, choosing one can be a bewildering business. Too often, the decision comes down to making a choice between two equally-attractive programs, each with specific functions you'd like to have, but neither having all the characteristics you need. Before you make a final choice, consider *LeScript*.

LeScript is a full-function word processing system that operates on the Tandy 1000, 1200, 2000 and 3000, as well as the IBM PC and IBM XT. It requires a minimum of 128K internal memory, at least one disk drive and MS-DOS 2.0 or higher. In order to get a good understanding of the program, let's look at some of its functions and features.

The Screen Display

LeScript's screen display is different from many word processors. When the program is loaded, an initial screen is displayed that allows you to choose one of four different types of monitors: high resolution monochrome, low resolution monochrome, color RGB, or Tandy 2000 color or monochrome. Once an option is selected, the program displays the editing or main text screen.

Four status lines are displayed at the top of this screen. The first is the name field. Here is where the name of the document is entered or displayed if an old document is loaded from a disk.

On the second line are several fields that display pertinent information about the document. The width field shows the character width of the screen. Eighty is the default, but the width can be easily changed. Next is the words field, which displays the total number of words in the document, and is dynamically updated as text is being typed in. The line field follows. Here, the number of lines in the document is dynamically maintained. The last is the free memory field. The remaining free internal memory allowed for the current document is shown in this field, and is automatically updated. For example, if you have a 256K machine, the initial free memory space is 169,984 characters. This value is decremented as text is entered.

The third status line contains the search and replace fields. These are two separate areas with which you can either search for a string of characters within the document or search for one string and replace it with another. Both fields allow strings up to 28 characters in length.

The fourth and last status line is the column grid. Much like a typewriter, this grid marks off the number of columns across the screen. Every tenth column position is numbered: 10, 20, 30, 40, etc. A triangular marker moves across the grid simultaneously with the advancing cursor within the text itself, thus showing you the exact position of the text cursor at all times.

The four status lines are always displayed at the top of the screen, and

are unaffected by scrolling or movement within a document during editing.

All word processors utilize word-wrapping and so does *LeScript*. But, in addition, the program uses reverse word-wrap. In the reverse process, the program determines, after each space is entered, if the words from the beginning of the current line to the space can fit on the previous line, and if so, it will move them to the previous line and rejustify the text. This process is automatic and is an interesting extension to the common word-wrap function.

The program also formats the screen text in four different ways: left, right, both and centered. Left margin justification is the most common format, while right, though not often used, is also available. When both the left and right margins are justified, spaces are automatically added between words so the text is flush to both the left and right margins. The centering function adds the appropriate number of spaces to the left and right margins so the text lies in the middle of the screen. These functions are evoked with the embedded control codes: JL, JR, JB and JC.

Because the text screen uses only four lines, the remaining screen space of 21 lines is a large and convenient window for entering or reading your written material.

The Editing Functions

Many of the 74 available editing commands perform the typical operations expected of any good word processor: insertion, deletion, block moves, copying text, disk file operations, cursor movement and similar commands. But more interesting are the special commands of *LeScript*.

Having more than just a handful of documents filed on a disk can be somewhat difficult to manage, especially if you must remember the name of each file every time you wish to load it. For some word processors, it's necessary to exit the program in order to view the directory. Not so for *LeScript*. You can view the files of any directory while still in the editing mode. While the directory is displayed, you may load or erase any file, and then return to the document you were working on. The directory command is executed by pressing CTRL-D. Pressing the CTRL-Z will terminate any operation in progress.

In fact, the majority of the program's commands are two-key combinations, with the control key usually being the first. For example, CTRL-I is the command for inserting text and actually functions as a toggle switch, alternating between turning the insertion function on and then off. Deletion works in a similar fashion. The simplicity of the commands is a real advantage—you don't have to wade through multiple menu levels to perform an operation, as with some word processors.

Another powerful, yet easy, feature of the program is its capability to intermix text formats. For example, you can left-justify a paragraph, right-justify the next, center the third and have the last justified to both margins, all on the same page if you wish. Column widths can easily be varied and intermixed as well. The codes for these formats, as well as other functions, are embedded in control lines which must appear before the text they affect. Headers, footers and pagination are also easy to add, change or delete from any document.

Like many word processors, *LeScript* can super- and subscript, but in a rather

unique manner. When a character is to be superscribed, CTRL-1 is pressed while the cursor is directly over the desired character. The superscript character then alternately blinks between itself and an up arrow to show it is a superscript. The same operation is performed for subscripts, except that CTRL-Z is used. The continuous blinking is a fascinating technique.

A similar display technique is used for italics. When the CTRL-5 is pressed for italicized characters, the characters blink between themselves and slash symbols (/). When bold characters are indicated, using CTRL-3, the characters are displayed in high intensity on monochrome monitors and in a contrasting color on RGB monitors.

For word processing, screen displays in color are often needless features, because most printers print in black. *LeScript*, however, uses color as a means of displaying special printing functions as described above, not just for cosmetic purposes. This is one way in which the program transcends most of its competitors.

Another useful feature is the help command — CTRL-?. When called, the text screen is replaced with the help screen — a list describing the editing functions and their codes. So, if you are in the middle of a document and want to perform an operation and have forgotten the command, the help screen is always a few seconds away; it's not necessary to look up the operation in the manual or on the quick reference card.

A variety of extended character sets is available with the program. Using the ALT key, in combination with other keys, you can generate European, mathematical and other special characters. These characters will be displayed directly on the monitor, but your printer, of course, must be able to print them.

The most powerful function of the program is its ability to work with other types of files. *LeScript* can edit BASIC files as well as *VisiCalc* data files saved in ASCII format. Almost every *LeScript* function for normal document files is available for use with these special files. If you're a programmer, this capability is indispensable.

For those of you curious about spelling checkers, the program has the capability of being integrated directly with *Electric Webster*. By simultaneously pressing the SHIFT, CONTROL and up-arrow keys, the program will automatically invoke the spelling checker and begin checking the document currently in memory. Naturally, you must have the *Electric Webster* program on the disk to use this feature.

To round out the editing functions, a few of the many other special features are automatic and semiautomatic hyphenation, an alternate text screen for editing two different text files at the same time, previewing a document in near final form before printing without control characters being displayed and the availability of a printing queue for up to twenty different files.

Printer Commands

Since word processing is nearly worthless without a printer, it is important for any word processing program to accommodate a variety of printers. In this arena, *LeScript* is difficult to beat, for it has 71 different printer drivers. Each driver is invoked by simply inserting its code in a control line of the document. The code is a K followed by the number of that particular driver. The manual lists the various printers and their driver numbers. The list of supported printers is impressive and includes almost every popular dot matrix and daisy wheel printer currently on the market.

"Word processing is one of the most important applications for microcomputers . . ."

The program assumes, by default, that a printer is connected to the parallel port of the computer, but if you have a serial printer, the program can be easily reconfigured to output the document to the serial port and also allows you to select nine different Baud rates, from 110 to 9,600 for MS-DOS computers.

In addition to the usual printer control functions such as line spacing, indentation, individual sheet pausing, emphasized print, text and sheet length, there are special functions that make *LeScript* a highly versatile word processor. Character pitch options (ranging from 5.0 to 17 characters-per-inch), character density options for proportional spacing and codes for printing selected portions of a document are all additional options. But perhaps the most useful feature is the printing of form letters.

Form letters are usually any text of which several copies must be printed and where a few phrases or words must be changed for each copy. Normally, form letters will have the same general text, but the addresses will be different. A typical application would be the sending of announcements or advertisements using a mailing list. For *LeScript*, the form letters feature works in conjunction with two separate files. The first is a text file that contains the general text itself and the second is a data file from which the variable (changeable) words or characters will be taken and inserted in the general text. The data file must be created in a specified format so that the records are properly identified. In the text file, the insert positions of the data file records must also be properly identified and aligned. Although this seems complicated, it isn't. The two demonstration files on the *LeScript* program diskette, along with the manual instructions, make the process painless. Once the two files are created, you can use the preview function to correct any errors before the letters are printed. The speed of the whole operation, once it is running, is dazzling.

Macros

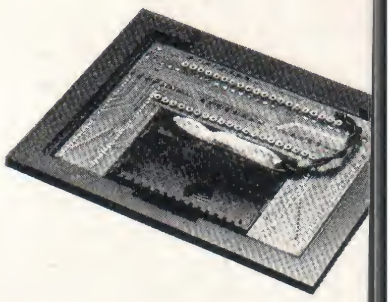
The last major feature of the program is the macro key capability. A macro is a list of predefined character strings that can be called and inserted in a document with a single keystroke. Say, for example, you have a series of phrases or names which you frequently use in your documents. Instead of typing them in each time they are needed, you can store those phrases or names in a macro and assign them to a specific key. Then when you want to insert that information, press the key and the characters will automatically appear, exactly as you had defined them. Obviously, this is an efficient technique for use in a variety of documents.

With the Tandy-style keyboard, having 12 function keys (F1 through F12), the program will allow you to define up to 60 different macro keys, with no upper limit as to the character length stored in each macro. The total of 60 is obtained by combining the function keys with SHIFT and ALT. With the IBM-style keyboard, having 10 function keys (F1 through F10), the program

Give your Model 100

128K RAM

Installs as easily as plugging in a socket



Software included, transfers from bank to bank. Works like main menu! Includes powerful RAM Basic that lets programs store and access data from any other bank.

PCSG says: Satisfaction guaranteed or your money back within 30 days

As amazing as it seems you can upgrade your Model 100 to 128K of RAM in just 60 seconds.

It comes to you right out of the box looking just like the picture. You just open the little compartment on the back of your Model 100 with a quarter and it just pushes right into place. You can then put the cover back in its place.

You then have 4 banks of RAM of 32K each. The additional three banks also work just like your Main Menu.

You push a function key and you are in the second bank. Push again and you are in third, again, then fourth. Press it once again for your original bank.

It has its own built-in NiCad battery that recharges right from the Model 100 and its guaranteed for a full year.

What is really great is that you can copy a file from one bank to another with just a function key.

Each bank is like having another Model 100, and all the built-in programs as well as any snap-in ROM programs appear in all four banks and work the same way. Your widebar cursor moves from file to file and you access any file or run any program just by pressing ENTER.

What lets you copy any file from one bank to another is a snap-in ROM from PCSG called RAM+, that comes at no extra charge. It just pushes right into the little socket in that same compartment with the 96K expansion unit.

Not only does this firmware let you copy a file from bank to bank, but you can make a copy of any file within the same bank instantly with a function key. Great for Lucid spreadsheets!

Copy a file from bank to bank with a function key

You can also rename a file, or kill any file with just a function key. Plus you can do a whole lot of other useful things like setting the date, day and time with function key ease. You even have a function key that lets you use non-Radio Shack printers without having to make those tricky dipswitch settings.

RAM+ lets you cold start any one of your banks without affecting the other three. That means that anytime you want you can clean out a bank's entire memory, but leave intact all the files in the other banks.

What is also fantastic is that you don't have to have the ROM in place to use the additional RAM. Whenever you take out the snap-in ROM it leaves behind a tiny machine code program that lets you switch from bank to bank just by pressing ENTER.

This lets you use your ROM socket to snap-in other ROMS like LUCID spreadsheet, WRITE ROM text processor, or DISK+ ROM file transfer program, and use them in any or all four banks. All of these, by the way, are available from PCSG.

When you are ready to copy a file from one bank to another or use any of the other fantastic functions we talked about you can just snap the RAM+ ROM back into place.

Everybody that has this 128K system in their Model 100 is so excited, because it gives them four times the capacity and all banks work just like the Main Menu.

And what has made a lot of people happy is that the system bus, located in the same compartment, is left free for you to plug in a DVI or the Holmes Engineering PCSG portable disk drive.

The ability to copy a file from bank to bank instantly with a function key, plus all of the other features make this RAM extension truly an engineering masterpiece.

Some people hesitate when they think of installing something, and then others are skeptical that any additional hardware could be as good as the Model 100 itself. That's why we sell these 96K expansions on a 30 day trial. Simply return it within 30 days for a full refund if you are not satisfied. Priced at $425. MC VISA COD.

1-214-351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. No. 207, Dallas, Tx. 75229

VISA / MASTER CARD / AM. EXP. / C.O.D.

allows a total of 50 macro keys. In either case, the storage space is quite sufficient for most word processing applications.

Another significant point is that you can also program *LeScript* editing functions into the macro keys. Therefore, you can perform one or a series of commands with a single keystroke. Furthermore, you can intermix editing functions and text. It's obvious that the programmable macro key feature is a powerful and captivating aspect of the program.

Without question, *LeScript* is a excellent word processor that performs its myriad of functions flawlessly. It is also a system which is easy to learn and use. The manual is succinct. The tutorial file provided on the program diskette is brief, to the point, and clearly demonstrates many of the editing functions.

One would be hard-pressed to find a superior system. So if you are in the market for a word processor, you should consider *LeScript*.

(Anitek Software Products, P.O. Box 361136, Melbourne, FL 32936, $199.95)

— Ralph Rideout

C-Num & *C-Sort*: Filling the Holes of Model 100 BASIC

When I first opened the manual to my newly acquired Model 100, I remember how impressed I was with the completeness of the portable's BASIC. Imagine having access to such high-level commands as CSRLIN and INPUT$ on such a tiny machine! I didn't even have these in the TRS-80 Model III BASIC with which I was weaned.

Unfortunately, my admiration turned to annoyance the first time I needed to renumber a BASIC program. It seems that Tandy forgot one of the most-used and necessary utilities a BASIC programmer can have — renumbering. I had taken for granted that renumbering was simply a part of the system as it is in most other variations of Microsoft BASIC.

Queue Software Systems has come to the rescue with *C-Num*, a machine language renumbering utility from their TRS-80 Model 100 Portable Computer Utilities Library.

C-Num works just like the standard BASIC renumbering utility, allowing you to start at any point in a program and assign new line numbers in the desired increments. During the renumbering process, all GOTO and GOSUB references are also changed to reflect the new line numbering. *C-Num* also performs error checking and informs you if there are references to non-existent lines in your program or if the new numbering creates line numbers less than zero or greater than 65,529. Other error checks look for increments less than one, line length greater than 255 bytes (in ASCII format) or insufficient memory to accommodate the new numbering.

Getting *C-Num* up and running is a simple process of loading a self-executing loading/relocating program from the *C-Num* cassette. This BASIC program automatically clears memory for *C-Num* and asks whether you want to load to the normal location at the top of memory. If not, you are prompted for

a new start address. If you don't know a start address from a street address, you just answer the first prompt with a 'Y' and the program is loaded to the default location.

The loader then converts an ASCII file on the tape to machine language and saves the resulting program, C-NUM.CO, to RAM. After that, executing the program is a simple matter of placing the cursor over C-NUM.CO and pressing ENTER. A copyright notice will briefly fill the screen and then you are asked for the name of the program to be renumbered. The target file has to be a BASIC file and the names of all BASIC files in RAM are displayed. You then type in the filename, the first line to be renumbered and the increment. Renumbering a lengthy program can take a minute or so, but small programs take only a second or two. If any errors are detected, the computer beeps and the line number and error are displayed on the screen. If there are no fatal errors, you are taken back to the main menu. Listing your program, you find that it has been totally renumbered.

For those of us who have painfully gone through and manually renumbered programs on our portables, *C-Num* is a blessing. The only addition I would like to see is the ability to renumber middle sections of a program. As it is, everything after the first line number you indicate is renumbered.

Another function I had gotten used to on my trusty old Model III was the built-in CMD"O" sort of TRSDOS 1.3. Since the Model 100 lacked one, I have had to use BASIC routines such as the one on Page 202 of the owner's manual. The problem is, BASIC routines require a lot of RAM, something very precious to most Model 100 owners.

C-Sort is the answer provided by Queue Software Systems. It is a machine language variation of the Shell-Metzner algorithm familiar to most BASIC programmers. Rather than sorting an entire file and writing the new one to RAM, *C-Sort* sorts three-byte pointers assigned to each record in the data file. The sorted results can be written to a RAM file, displayed on the screen, dumped to a printer or saved on tape. This method saves considerable memory and allows files to be sorted which would be too large using a standard BASIC routine.

This program is ideal for use with the Model 100's built-in appointment and address book programs and appears to have been designed specifically with them in mind. *C-Sort* assumes that all elements in a record will be on a single line with no carriage returns. You can specify where on the line you want to begin and end the sort, which is almost instantaneous. Unfortunately, it doesn't work too well with other data formats. I tried everything I could think of to use it with a program I had written. My program wrote each field in a record to a separate line so that the data file could easily be uploaded to a Tandy 1000 and appended to a similar program's random access data file of the same format. It just wouldn't work. I suppose it was a bit much to expect, but it shows that *C-Sort* is not as flexible as it could be.

Getting this one from tape to RAM is a little more difficult, but still fairly simple. You first load C-SORT to BASIC from cassette and then save it to RAM. Then you type CLEAR 0,62537 and CLOADM"SORT" to set the start address and load the machine code from cassette to RAM. Next you type SAVEM"SORT",62537,62908 to save SORT to a RAM file along with the starting and ending addresses. At this point, you can run the program by placing the cursor over C-SORT.BA and pressing ENTER.

The installation might sound a bit confusing to those who are not used to working with machine language programs on the Model 100, but the manuals for *C-Sort* and *C-Num* hold your hand through it. Although the manuals are pretty skimpy-looking, they do provide clear instructions for these easy-to-use utilities. It would appear that phone support is not something Queue wants to offer, though. I couldn't find a phone number for the company listed in either manual or in any advertising.

C-Num and *C-Sort* are well worth their prices and offer handy additions to any Model 100 library. *C-Num* is an absolute must for BASIC programmers and *C-Sort* is a useful tool for all 100 users.

(Queue Software Systems, 4528 Belleview, Suite 210, Kansas City, MO 64111; *C-Sort* $24.95, *C-Num* $19.95)

— Richard Burckhardt

Software **1000/1200/3000**

The Newsroom, A Tool For Amateur Publishing

The Newsroom is impressive. It has a nice combination of clip art, graphics, word processing and communication capabilities. The original package comes with 600 pieces of clip art and an additional 600-piece volume of clip art is also available. Graphics tools for use with the keyboard, a joystick or a Koala pad, allow modification of the clip art provided or the creation of new clip art. The word processing features allow the use of five type styles in coordination with the clip art to produce a newspaper-type document of your own design.

The wire service (communications) features allow the transfer of data among IBM PC/PC*jr*, Commodore 64 and Apple II series computers. This transfer capability can handle both the graphics and text. Your newspaper can be started on one type of computer and transferred to another type of computer for further work. A group of people at different locations around town can all work on the same project even though they each use different equipment.

The Newsroom is copy protected in a way that allows the files on the *Newsroom* disks to be backed up. The backup disks cannot be run themselves, but they can be used to restore any files on the original disks if necessary. *The Newsroom* comes with a master disk and a second disk for the clip art library. The additional, extra cost, Volume 1 of clip art takes one disk. The documentation and online prompts step you easily right through the procedure to get set up for your equipment configuration. Procedures are provided for both one-drive and two-drive systems. The IBM setup procedure contains special options specifically for the Tandy 1000 and 1200. *The Newsroom* requires 128K and DOS 2.0 or higher for the IBM compatible machines.

A large variety of options for popular computer printers are included with *The Newsroom*. The printed outputs are designed around either 8½ by 11-inch letter-size paper or 8½ by 14-inch legal-size paper. The output page may be divided into panels and banners. The

banner is the full width heading on the first page of a document. It amounts to the top quarter of the page for letter size and the top fifth of the page for legal size. The rest of the first page has two columns of panels, three panels in each column for letter size and four panels in each column for legal size. The banner doesn't need to be the first page, it doesn't even need to be used. I refer to it being part of the first page because that is the most common way for it to be used. Continuation pages normally have two panels, one in each column, in place of the banner.

Letter-size pages can have either a banner with two columns of three panels each or no banner with two columns of four panels each. Similarly, legal-size pages can have either one banner with two columns of four panels or no banner with two columns of five panels. The banners and panels are only used to break the pages down into workable pieces. The end product is combined as though it were all one piece with no visible separation of banners and panels. The panels and banners are just an imaginary breakdown for working on your paper one piece at a time. A sentence can start on one panel and finish on the next.

Actually, a full page doesn't even need to be produced. A panel by itself, a banner by itself or even a partial page of data can be printed without any problem. Even the clip art (photos) can be printed without being placed into a panel or banner.

Now that the normal layout of a page and the setup procedure have been explained, let's get on to using *The Newsroom*. When first fired up, a graphics display of the six main menu items is shown. The six choices are Photo Lab, Press, Wire Service, Banner, Copy Desk and Layout. Each of these will be covered separately. The menu choice can be selected by use of the arrow keys on the keyboard, a pencil movement on a Koala pad or the movement of a joystick. As the cursor moves around the main menu, the cursor location is indicated by highlighting with bright video. When the proper choice is highlighted, the selection key on the joystick or Koala pad is depressed to make the selection. For using the keyboard, the ALT key is depressed. This same selection procedure applies to all menus and the graphics tools of *The Newsroom*. For this article, the selection key refers to the Koala pad or joystick selection key unless you are using a keyboard. If you are using a keyboard, the selection key refers to the ALT key.

The use of a keyboard moves the cursor around the screen in small jumps. This is not ideal for working on fine graphics detail so an option was included in *The Newsroom* for moving in very small increments. By simply depressing the shift key while using the arrow keys, the keyboard movements are reduced to the very small moves needed for detail work.

Now for the first main menu choice. The Photo Lab on the main menu has an illustration of a photographic darkroom. The Photo Lab has its own menu of choices with icons down the left side of the screen. A work area takes up the remaining screen area. The icons which can be chosen are these: a picture to allow the selection of clip art from a library disk, a left and right arrow to allow reversing (or flipping) the clip art in the work area as in a mirror image, a crayon to allow calling in the graphics tools window, a magnifying glass to

blow up part of the work area for detail work, the word "oops" to undo what was just done, a trash can to clear the work area, a camera to take a snapshot of the work area or a portion of the work area for saving to disk, a disk to allow saving a photo (from the snapshot) to your data disk or loading an existing photo from your data disk, and the word "menu" for going back to the main menu.

To build a graphics image from a clip art library, the picture icon is selected to call up the clip art library. For a two-drive system, the master disk is in Drive A and the clip art disk goes in Drive B. As is the case throughout *The Newsroom*; if the proper disk isn't there already, the disk is asked for with a polite message. When the disk is available, an index of the clip art is displayed which allows the selection of a particular set of clip art by name. The same clip art set names are listed in the documentation with illustrations. The selected clip art set is displayed on the screen when selected. A hand appears on the screen in place of the cursor. Move the hand onto the particular piece of clip art you want and use the selection key to select it. Once selected, your photo work area appears on the screen again with the clip art included. Use the hand to move the clip art around within the work area with the arrow keys, the joystick or Koala pad pencil. When using a keyboard, remember to hold the shift key down to move the clip art in small increments for final detail work. When properly located, the clip art is dropped by depressing the selection key again.

At this point, you can call up the graphics tools, flip the image, magnify the image for detail work, clear the work area to start again or define an area of the work to be saved with a snapshot. After saving the clip art to disk, the finished photo can be used again. You might want to add text to the photo itself or clip the photo by use of the camera or even add your own touch to the art by use of the graphics tools. Normally the photo would be loaded from your data disk to become part of a panel for inclusion on a finished page of output. However, the photo can be printed by itself, transmitted via the wire service or just saved for possible future use.

The Banner selection from the main menu offers the same options as building a photo except that there is no camera icon for snapshots. The main menu shows a person working at a drawing table with "Banner" for a heading. The banner is double the width of a full photo or panel in that it covers both columns of a page across the top. Graphics for a banner are formed by the same methods as in a photo except that it isn't necessary to take a snapshot. Previous banners can be loaded from a data disk for modifying, transmitting, or printing. New banners can be created using text, clip art, your own art with the graphics tool box or any combination of these.

The graphics tools include choices for line, lines, circle, box, draw, erase, 10 pen sizes/styles, 10 fill patterns, the hand to move images around and five text types. The text choices are small serif, small sans serif, large serif, large sans serif and large English. The graphics tools are about equivalent to some other software packages which offer graphics tools alone without any of the other features of *The Newsroom*. It may be that a stand-alone graphics package would typically offer a few more graphics features (such as color choices), but *The Newsroom* is quite complete in itself.

"Anyone can have fun with it. If you want something to really catch the interest of your children in creative writing, get *The Newsroom*."

The Copy Desk from the main menu is for building or modifying a panel. This choice is indicated by an illustration of a news person at a desk with a typewriter. The Copy Desk has icons for font, erase, oops, trash can, disk and menu. The same font choices apply here as mentioned earlier about the graphics tool box. The Copy Desk menu choices are similar to those for banner or photo work. There are special features within the Copy Desk however. The erase option clears all text from the work area, but retains any photos. The trash can option clears the entire work area. Photos as defined in the Photo Lab can be loaded from your data disk for inclusion in the panel. This is normally the first step because the entry of text will automatically flow around any photos only if the photos are there first.

The Copy Desk is the area which has the most word processing features. Note that only one small and one large font can be used in any one work area. The moment you switch from one type of small font to another, all of the small font in the work area makes the same change and similarly for large fonts. Basic word processing features are included. Insertion within existing text is automatic at the cursor location. The delete key is used to delete small amounts of text. Text can be deleted or moved in blocks by first marking the boundaries of the block with the selector key. CTRL-Q and CTRL-W are used for moving and copying text. It is really neat the way the text will automatically wrap around any photos in the work area and the way the text fits together from one panel to the next on the final printed output.

When a panel is finished, you normally save it to your data disk by selecting the disk icon. Another panel can easily be started by using the trash can option to clear the work area before starting another panel. When all panels are completed, the next logical step is the Layout choice of the main menu for assembly of pages. The main menu illustration for Layout shows a cut and paste operation. Of course using *The Newsroom* is much easier than any cut and paste operation with real glue. Who needs sticky fingers!

The first choice for Layout is responding to a prompt about laying out a page with a banner, laying out a page without a banner, loading a previously saved page from your data disk, or returning to the main menu. Layout is quite easy. After you select one of the page layouts, you just move the cursor to the banner or panel you want to work on and depress the selection key. If you were in the banner area when you depressed the selection key, the banner names will be displayed as you defined them when they were saved. You just move the cursor to the banner name you want and depress the selection key.

The next display will show a layout of the page with the name of the selected banner in the banner area. Then you just move the cursor to the panels one at a time and select the previously-saved panel names for inclusion in the page. If you change your mind, you just select a new banner or panel name for inclusion on the page. When you finish the

page, the save option is selected to save the entire page to your data disk with whatever name you choose. Build as many pages as you wish.

To see what the actual printed page looks like, just go back to the main menu and select the Press option with the illustration of a printing press. In the Press you have the choice of changing the printer setup to match your equipment, printing or going back to the main menu. A page, panel, banner or photo may be printed. Normally the finished product will involve printing a page which may include a banner and which will include one or more panels. You just select the print page option and then select the page name which you want to print. Here again, as throughout all menu choices, *The Newsroom* will prompt you for the proper data disk if it isn't found in the drive already. You are also reminded to have your printer ready before the printing starts.

The printing is similar to a graphics screen dump. To me this is the only significant problem with *The Newsroom*. The print quality isn't really up to the standards of most people that are publishing a professional looking newsletter, newspaper or similar type of document. The print quality is certainly readable, especially so if you have a good ribbon in the printer. On the other hand, there is no choice about double-strike or near-letter-quality printing. In fact, there just isn't any choice about the print quality. I imagine the print quality is fine for some small informal newsletters and for classroom work on how to put a newspaper or newsletter together. Children should also love it for creating customized stationery by use of the banner option or for creating nice graphics printouts.

Perhaps I am being overly critical here. I have seen some articles in computer journals about professional publishing recently. That normally involves a laser printer, a loaded microcomputer and software costing many thousands of dollars. The list price for *The Newsroom* with the basic system including 600 pieces of clip art and with one additional volume of 600 more pieces of clip art lists for about $90. Unfortunately, I don't have a laser printer at my disposal for trying *The Newsroom* with that type of printer. A laser printer might not even be one of the printer choices that comes with the system. *The Newsroom* just isn't aimed at that market, it's more for home and school use and meets that need very well.

Another problem with *The Newsroom* is that the wire service doesn't allow changes to the communication parameters. I couldn't use the wire service because my modem doesn't send the standard CONNECT string, etc. I don't consider this a significant problem for most people because most people wouldn't get *The Newsroom* for its communication features anyway. However, this could potentially be a significant shortcoming in some special situations.

The documentation manual for using *The Newsroom* covers all of the features in only 44 pages. That is approximately half of the manual; the other half is devoted to a guide about how to create a newspaper. I am truly amazed that only 44 pages are needed to cover all of the features for multiple types of computers using the keyboard, a joystick or a Koala pad. The menu choices and the prompts are so well laid out that this small manual won't even be used very much. Everything is done in a very

logical way that anyone can follow. The only pitfall here may be forgetting to save your work before moving on to the next step. My 9-year-old daughter was using *The Newsroom* without any trouble except that she forgot to save some of her work. Perhaps a future release of *The Newsroom* will have more reminders about saving your work before moving on to the next step.

The documentation covers everything and is well-written. It's a compliment to say that everything is well covered in only 44 pages. The documentation recommends separate data disks for photos, panels, banners and page layouts. You can use one disk for all of these if you wish, but you may run out of room on the disk once you get started trying all the handy features and options of *The Newsroom*. The documentation is packed full of samples for building newspaper banners with clip art and for using clip art within the text. It is a well known fact that clip art adds a great deal of comprehension to any written document.

The Newsroom is truly a fine product and well worth the cost. Anyone can have fun with it. If you want something to really catch the interest of your children in creative writing, get *The Newsroom*. If you want to create your own newspaper or newsletter and aren't too concerned about professional print quality, get *The Newsroom*. You probably wouldn't use a printer at home if you were too concerned about print quality anyway. *The Newsroom* would be a fine addition to your selection of software.

(Springboard, 7808 Creekridge Circle, Minneapolis, MN 55435, $59.95 for *The Newsroom* and $29.95 for *Clip Art Collection Volume 1*. Includes a 30-day money back guarantee.)

— Robert Jensen

Software **1000/1200/3000**

It's a *Miracle*

Well, perhaps it's not a "miracle," but it is certainly a prodigious help for "power" database users.

This is a top-quality piece of software with documentation that is a work of art from the visual standpoint.

Miracle is also very complex or, at least, it can be very complex.

Miracle contains a very usable wordprocessor which includes such things as automatic date insertion and (if you have over 256K memory) up to four windows. (*Miracle* windows, by the way, aren't portions of the larger text or spreadsheet, they are independent worksheets.)

The windows are made easier to use by the optional use of different color background and print in each window (on color graphics-equipped computers only, of course).

While working with up to four different wordprocessing files at once (either displayed simultaneously or with just one zoomed to fill the screen), you can cut and paste between these different documents.

When using Sheet, the *Miracle* spreadsheet (256 by 64), you can also open up to four windows, each inde-

pendent, with separate information. You can also build a three-dimensional spreadsheet by linking up to four separate sheets giving you a true 3-D spreadsheet.

For example if you have one sheet named "first" and another named "second," you can either use them as two completely independent spreadsheets, or call information from "second" into "first" using a formula entered into some cell in "first."

If you already have spreadsheets in use, *Miracle* provides for conversion of *Lotus 1-2-3, VisiCalc* and *Multiplan* files to *Miracle*'s spreadsheet along with a conversion program for changing ASCII files to *Miracle*'s document files for the *Miracle* wordprocessor.

You can also set up a simple "plus/minus" graph right on your spreadsheet or, if you have color graphics, you can call up a full screen view of any of a number of two- and three-dimensional graph forms defined to your current spreadsheet.

You can also (if you will settle for a mundane four-color bar graph) have a graph displaying the relation of selected ranges of spreadsheet elements on the right third of your spreadsheet page. This graph, of course, changes as you change data in the spreadsheet. Let me make this clear; the graph actually shares the screen with the spreadsheet if you wish.

The graph types include: area, bar (two- and three-dimensional and stacked), line, pie (two- and three-dimensional), X-Y and a special stock trend analysis chart (high/low/close/volume). You can switch between various types of graphs in a few seconds to permit you to select the one that most clearly presents your data.

The real miracle of *Miracle* is the way all of these functions are tied together. Using Agenda you can program *Miracle* to call up a data base, log on, download files to a spreadsheet and graph the information — all automatically.

The Comm module supports "XMODEM" and "XFER" protocols, along with the transfer of ASCII files.

Miracle will also accept calls (unattended) from other computers, so be careful who you let it talk with.

Comm supports up to 2,400 Baud modems (Hayes compatible or any command driven model) and up to 9,600 Baud direct (XON/XOFF). (I was not equipped to verify this beyond 1,200 Baud but it does work with Tandy's 300/1,200 Baud modem.)

Miracle accepts (for each file) a primary and alternate phone number and you can specify up to 99 retries each. After a total of 198 tries (maximum to your specification), *Miracle* gives up.

Miracle comes on line emulating a TTY terminal but will also emulate Digital Equipment's VT-100 and VT-52 terminals (although not completely).

Miracle will accept 10 macro keys, each with 64 characters, but you don't need to use these for phone numbers and log on procedures. These are contained in `.TRM` (terminal) files.

For each database you set up a file called `SCRIPT`. `SCRIPT` contains things such as your terminal identifier, user ID, password and the instructions to take you into the part of the database you wish to use. Building a `SCRIPT` file that recognizes system prompts and responds correctly could be quite a job with a text editor (although I did it in a few minutes for Delphi), but you don't need to go through that. If you use *Miracle*'s learn mode, *Miracle* can build a `SCRIPT` file while you access the database in your normal fashion — *Miracle* memorizes the procedure as you perform it.

Each set of instructions for different databases (including protocols, phone numbers, macrokeys and script files) is saved in a separate terminal file that is presented for your selection at the beginning of each session. If you choose one of the existing files, you are automatically logged onto that system (if that is what your `SCRIPT` file calls for).

Compatability is always a problem and *Miracle* includes translation tables to help. The translation tables (one for each data path: display, terminal, capture buffer, printer, etc.) are filters that allow you to specify the output for each individual ASCII character separately. If you want ASCII code "049" to produce 'A' instead of '1', just change the character in the appropriate table.

Documentation

My only real complaint about *Miracle* involves that beautiful documentation. I first tried to use the program on a 1200HD with 256K and no modem. Portions of the program worked but not as the documentation said they should. In particular, I could not access the terminal program to prepare files for the arrival of my modem. In addition, I could not open windows in the wordprocessor and could only barely open

one in the spreadsheet. The people at Micro-Systems explained that, quite reasonably for a communications program, *Miracle* would not permit access to the terminal program without an installed modem (or RS-232). I was also told that windows wouldn't work in 256K.

Every program has its system requirements and there is certainly no fault with *Miracle* because it won't work without required hardware. My complaint is that two lines explaining these exact restrictions in the documentation would have saved my trying to reconfigure my system by modifying `CONFIG.SYS` and several attempts to reinstall the software — not to mention a call to Florida. I couldn't find anyplace where the literature or documentation mentioned that you really need more than 256K to make full use of the program. (Speaking of customer support, the people at Micro-Systems were very helpful — even before they knew I was writing a review.)

Another curious lack in the documentation was in the spreadsheet instructions. Nowhere in the instructions does it mention how to enter a custom formula into a cell (you use '=' before the formula).

I guess I am upset with the documentation because the rest of the program is so terrific. Did I mention that there are 185 screens of information available at the Help command? This onscreen help is so large that it has both an index and a table of contents. In the help section for the spreadsheet the built-in functions are explained and there is even an on-screen demonstration using a (static) portion of the spreadsheet to illustrate proper use.

Conclusion

Miracle is a well-designed, easy-to-use and very sophisticated integrated communications package. With an enhanced computer system (expansion memory and graphics) it is also an attractive looking program with advanced wordprocessing and spreadsheet features (3-D spreadsheets with on-spreadsheet graphics and multiple, independent windows on both) but, as Micro-Systems Software's own documentation points out, this is primarily a communication program.

As an active investor, I can see that using *Miracle* to update my spreadsheets and graphs automatically will save enough on-line expense to pay for the program in a reasonable time.

One important note; the 1200 version of *Miracle* differs slightly from the IBM version, so be certain you get the correct version. Since *Miracle* is available through Radio Shack's Express Order system, ordering through Radio Shack is the easist way I know to be certain of getting the correct version.

There are other programs available with similar features, but I know of no other that can offer this ease of use for such a reasonable price.

Miracle does not permit direct-to-disk saves of captured data (the program saves to a volatile memory file which can then be saved to disk). The people at Micro-Systems tell me that *MTERM II*, another communications package available from them, will download directly to disk.

I would like to say a special word of thanks to the people of Perry Computers without whom I would not have received a modem in time to complete this review.

(Micro-Systems Software Inc. 4301-18 Oak Circle, Boca Raton, FL 33431, $299. EOS # 90-0293)

— John McCormick

TMPC Manages Your Time

TMPC stands for "The Most Precious Commodity" time. The program by that name, developed by Acroatix Inc., does an excellent job of helping you manage your time, keep track of your assignments and appointments. It even makes you rate assignments according to both priority (on a one-to-three scale) and stress (on an A-to-C scale). Its instruction manual is well-written and easy to understand. It contains a thorough tutorial and a summary of all program functions, all written in plain English.

Using *TMPC* is simplicity itself. The program is divided into 14 screens through which you move a person-shaped cursor to enter and review assignments and appointments. On the way, the program offers many little touches — one screen even allows you to change the shape of the cursor if you

find the tiny person too cute for your taste.

The screens are arranged in a logical, three-column tree. The left-hand column contains three screens for assignment processing. These allow you to add new assignments, which the program calls "ToDos," and rate them for stress and priority; view the assignment list; and process individual assignments (either hold, cancel, or move them to the list of things to do today).

The central column deals with today's status. From the top, the screens are: the date, overall status (the number of assignments and appointments listed as upcoming), daily status (the number of assignments and appointments on your plate for today), and "Process To Do" where you mark individual assignments as done. Two additional screens are available from the "Process To Do" screen. To the left is "Record Functions" which allows you to change the cursor's shape and speed through *TMPC*'s corridors. To the right is a future expansion screen that will allow you to use expansion modules that Acroatix is developing.

The right-hand column allows you to set and view appointments. Its three screens include the calendar, which allows you to select a date other than today; "View Appointments," which shows the appointments and memos attached to the date you have selected (today if you have not used the appointment calendar); and "Add Appointments." This bottom screen has several conveniences. You can attach a warning to an appointment, for instance, and specify how many days ahead you want the warning to appear. These warnings appear on the "View Appointments" listing. It also allows you to set up weekly and monthly repeating appointments. Therefore, if you have a deadline on the 15th of every month or a meeting every Wednesday you only have to enter it once and use the monthly or weekly repeat screens to have it automatically repeated. The monthly repeat screen even lets you program *TMPC* to move the appointment ahead or back to the nearest business day if it happens to fall on a weekend or holiday in a given month.

Unfortunately *TMPC* lacks an annual repeating feature, so holidays, birthdays and anniversaries must be manually entered. Entering appointments is a tedious process, requiring you to move back and forth through three screens. *TMPC* does not display any unified list of the assignments and appointments you have to do today. You can, however, print the list of your assignments for the day using a command in the "Process To Do" screen. *TMPC* also does not allow you to search for an appointment when you can't remember its date. The only way you have of finding it is to consult the appointment's calendar and check dates that are marked as having appointments on them one by one.

All of these things can be fixed with additional modules. *TMPC* unfortunately suffers from a much larger problem that modules will only exaggerate. It takes up too much room. *TMPC* is a machine language program that takes up 9,600 bytes. The file it sets up to hold your data adds several hundred more. That is a lot of overhead in a machine that has a maximum of 32K memory. Nor will you want to take *TMPC* out of the machine when you aren't using it — since it is your appointment calendar, you will want it available at all

times in case you need to make an entry. Furthermore, as an assignment tracker it is only effective if you can check off assignments and select new ones as you complete each job.

Time might be your most precious commodity, but if you are to get full use out of your Model 100 the byte count is also vital. *TMPC* does not do anything that you cannot do just as easily manually with an appointment book. Whether you can afford the space depends on what else you use your Model 100 for, but you should seriously consider the hidden cost of computer memory overhead before investing in this otherwise excellent program.

(Acroatix Inc., Box 273, Wilmington, MA 01887, $50 plus $3 S/H)

— G. Berton Latamore

Software **1000/1200/3000**

Infidel — An Adventure by Any Other Name

First let me get something straight— I am not a big aficionado of Adventure games. I'm not all that good at solving them either. Now that I've lost all of the Adventure purists because they know that I'm "one of those," I'll proceed.

When I heard of the "Interactive Fiction" series by Infocom, I was intrigued. I wanted to try playing an active roll in a novel (if you will), turning the plot this way and that depending on the decisions I made. That is why I wanted to try one of this series of programs in spite of the disclaimer I made in the first paragraph.

Infidel by Infocom is a thoroughly creative program. A computer system which runs MS-DOS 2.0 and uses 5¼ inch diskettes is required, with either a 40- or 80-character display. A Tandy 1200HD will do just fine. I spent well over an hour reading the literature which accompanies the non copy-protected diskette. This literature is designed to immerse the participant in the mood of the story which is about to unfold. Included are entries from a handwritten journal (written by your character) detailing how this great opportunity fell into your lap — the chance to head an archeological expedition to Egypt and gain the glory and respect you so rightly deserve — not to mention the untold riches you hope to unearth!

The offer comes from an elderly woman whose father explored the area in the early part of the century, discovering some tantalizing clues concerning the location of a lost pyramid before the hostile desert claimed his life. You have his map and a stone cube with hieroglyphic clues. You set off with these and a crew of locals to find the spot in the middle of the desert.

The journal proceeds to tell that all does not go well with your expedition. Equipment malfunctions, short supplies and cultural frictions bring the locals near the point of mutiny. You sit in your tent and decide to write an encouraging letter to your sponsor back in the U.S. The letter is also included with the program, and is quite humorous. Little do you know, but the refreshing native beverage which you sip while

you write has been drugged by your disgruntled crew. The letter trails off into some really hilarious delusions.

The program then begins: You awake, head splitting, to find that your crew has abandoned you after taking most of the supplies. You must attempt to stay alive. Managing that, your goal is to find the lost pyramid, make your way to the burial chamber, and uncover the sarcophagus, thus gaining respect and riches.

Infidel, by any other name is an Adventure game. Its responses to your actions sometimes fill a screen with entertaining, mood enhancing prose — in that respect it is different from any other Adventure game program I have encountered. *Infidel* also differs in the complexity of the actions which you can enter.

The program will accept whole sentences which combine more than one action or which include multiple objects for a verb. "Open the door then South then close the door then lock it" would be a perfectly acceptable set of commands for *Infidel*; as would "Put the torch and the map in the knapsack." You are no longer limited to actions like "Take sword" and "Stab dragon." *Infidel* has an extensive vocabulary so communication is more likely to be successful. If your command is a no-no, however, *Infidel* has an extensive vocabulary to get you on the right track, such as "You must supply a noun" or "I can't see any shovel here."

You can select from three levels (Verbose, Brief or Superbrief) just how wordy you want *Infidel* to be in its descriptions. The SCRIPT command will allow you to make a transcript of all of your actions with your printer. This is handy, but can use a lot of paper (especially if you select Verbose). SAVE will allow you to return to a certain point in the game after you die (and die you will!) without starting from scratch. Use SAVE early and *often*!

How did *I* do? In my first attempt I followed a mirage across the desert (well hey, I was thirsty!) to my demise. I did get to "see" a number of entertaining hallucinations along the way though. My second attempt was much better. I successfully attended to tasks necessary to my survival, explored my surroundings collecting necessary tools and items, located and obtained entry to the hidden pyramid, then promptly fell to my death on a stairway! Actually I was rather proud of my accomplishments.

Infidel did not, however, share my enthusiasm — it rated me a "fumbling beginner." I admit it — I get to a certain point in my expedition and I'm stymied. I even know what I need to do, but I can't quite figure out how to get it done. I see there is a book of hints available from Infocom — maybe that's the answer.

As Adventure games go, *Infidel* is first rate. My disappointment stems from my expectations that "interactive fiction" would be something very different from Adventure games. A notch above, I admit, but an Adventure game nonetheless.

(Infocom, Inc., 55 Wheeler St., Cambridge, MA 02138, $44.95)

— Stanley Townsend

Software **1000/1200/3000**

Wizardry — Proving Grounds

Want to go on a trip? I don't mean the type of trip to nice sunny places with smiling faces via the "Friendly Skies." This trip is to a dark, magical place inhabited by zombies, thieves, spell-carrying, evil-cursed creatures, and just to keep things interesting, treasures beyond your wildest dreams.

Instead of boarding the great silver Boeing birds, the means of transportation to this magical place is via your MS-DOS Tandy computer and a program called *Wizardry*.

Wizardry is an Adventure game, unlike any Adventure game I've previously played. It is a complex game, for one to six players, with all players controlling one or more characters who go off on expeditions in search of loot and glory. The more successful a character or group is, the more powerful they become. Each character is specialized. Some are good fighters, some can cast magical spells of many types, and some are good at defeating the many traps that guard the treasures. As a character becomes more powerful, he may gain some general abilities, but in all cases, cooperation is the key to success.

The first step in *Wizardry* is to create your cast of characters. A character is your "alter-ego." Like anyone, he has various skills, abilities and possessions. You control him by telling the computer what you want him (or her) to do. All the information about your character is kept on a scenario disk until you wish to use him, and after you finish a game, it is returned and updated. Thus as you continue to play with him over many sessions, the more you play, the more capabilities he gets, and the more challenging and interesting the game becomes.

Each character has five basic characteristics. These are:

- Strength — affects combat ability.
- I.Q. and Piety — determines the ability to cast mage (more on this later) and priest spells.
- Vitality — modifies the amount of damage that can be sustained before death.
- Agility — determines the order in which attacks occur.
- Luck — helps in many mysterious ways.

Based on the above five characteristics, your characters may become one of eight classes of players. Each class has certain minimum characteristic scores that qualify a character for that class. As you play, and your players become more powerful, they may qualify for a better, more powerful class.

The classes are:

- Fighters — The basic warrior who can use any weapons. The strong and dumb type, but handy to have on your side in a fight (and there are a lot of those).
- Mage — The sorcerer who can use magical spells and very limited weapons. While Mages at first don't seem to have much use, those spells are very handy to have.
- Priests — The holy men who do not fight as well as Fighters, but have the ability to dispel. Some monsters in the maze, collectively known as the "undead," can be forced back into the abyss from whence they came by a Priest's dispel.
- Thieves — They are very good at circumventing the noxious traps that may be between a party of players and some loot.
- Bishops — They are a combination of Priests and Mages, and have advantages and disadvantages of both. They also have the ability to identify the nature of magical items, thereby avoiding hefty charges levied for that service at the castle.
- Samuari — They are fantastic fight-

ers and can begin to learn magical spells at advanced levels. I'm not sure what Japanese warriors are doing in an English medievel game, but what the heck, it's best to relax and don't let logic cloud the matter.

- Lords — They are a combination of Fighter and Priest.
- Ninjas — They are inhuman fighting machines. Again the Japanese warriors? Possibly the programmer spent too much time in a sushi bar.

There are many other characteristics of the adventurers. Alignment (good, neutral or evil), race (human, elf, dwarf, gnome or hobbit), ability, age, wealth, equipment, armor, etc. These all relate to how the group fares as they wander through the game.

After the characters are created, your quest for glory and treasures is started. The Castle is your starting and ending point for all expeditions. There you can rest, create parties, get help from the gods, buy and sell equipment, or go to any other part of the game. After you have gathered together in a party, purchased or traded for the best equipment you can afford, and rested to restore hit points and spells, it's time to leave the safe, but expensive, confines of the Castle and enter the unsafe, but possibly lucrative, corridors of the maze. Here's where the real Adventuring begins!

Wizardry uses a 3-D low-resolution perspective plot of the maze as you would see it if you were actually there. You see the walls of the maze extending into the distance. By using spells (if one of your group can use spells — hint, hint), you can light up the maze so you can see farther and see hidden doors.

Without some sort of challenge, *Wizardry* would be just a game of mapping. Well, fear not, challenges abound in the form of monsters. Some are wandering around, some reside in rooms. In any case, since they generally do not like being disturbed by outsiders, and really do like the taste of human (or other) flesh, it is likely that a good many fights will occur. The outcome depends on how well-prepared you are and the actions you choose to execute at a given time. Luck doesn't hurt either.

The object of this Dungeons and Dragons-type game is to survive, gather as much treasure as possible, return to the Castle and then go out and do it again. As you complete a foray in the maze, your characters hopefully gain strength and ability and the next excursion can be more successful, going to deeper and deeper levels of the maze. Unfortunately, your characters may not always be so fortunate as to return alive. But, spells, magic and often large sums of money have been known to reincarnate many a player.

If your idea of a good relaxing time on the computer is a quick game of *Donkey Kong*, *Wizardry* is *not* for you. This is a game that requires your attention for hours at a time, time after time. It's best played in the company of fellow Adventurers who can share ideas and experiences to successfully return from the depths of the maze.

If this extended type of Adventure has been of interest to you, *Wizardry* is a well-programmed, well-documented example of an interactive fiction fantasy.

(Sir-Tech Software Inc., 6 Main Street, Ogdensburg, NY 13669; 315-393-6633, $59.95)

— Bruce Rothermel

The following products recently have been received by PCM, examined by our magazine staff and approved for the PCM *Seal of Certification*, your assurance that we have seen the product and have ascertained that is what it purports to be. This month the *Seal of Certification* has been issued to:

Adventure Alpha, a unique Adventure game in which classic mathematical and logic problems must be solved in order to survive and locate treasures. Requires Tandy 1000, 1200 or 3000 and graphics option. *Milliken Publishing Co., EduFun! Division, 1100 Research Boulevard, St. Louis, MO 63132, (314) 991-4220. $34.95.*

BASIC Programmer's Toolkit, a utility for BASIC programmers which will print formatted program listings and cross reference listings. Will also remove non-referenced program lines for more effecient compiling. Requires Tandy 1000, 1200, 2000 or 3000. *MVP Software, 1035 Dallas SE, Grand Rapids, MI 49507, (616) 245-8376, $39.95.*

The Black Cauldron, an animated Adventure game based on the Walt Disney movie by the same name. Requires Tandy 1000, 1200 or 3000 and graphics option. *Sierra On-Line, Inc., avaiable through Radio Shack stores nationwide, $39.95.*

The Cheapware Text Editor, a text editor to be used primarily for writing and editing programs. Requires Tandy 1000, 1200 or 3000. *Robert L. Nicolai, 4038 N. Ninth Street, St. Louis, MO 63147, $20.*

The Cheapware BASIC Program Mender, will read damaged BASIC program files and attempt to recover any lost portions of the program file. Requires Tandy 1000, 1200 or 3000. *Robert L. Nicolai, 4038 N. Ninth Street, St. Louis, MO 63147, $20.*

Chuckle Pops, a background program that, when called, will "pop up" jokes on the screen, even while running other application software. Requires Tandy 1000, 1200 or 3000. *Enlighten, P.O. Box 2037, Ann Arbor, MI 48106, $14.95.*

Cormail, prints mailing labels from *DeskMate* files. Requires Tandy 1000, 1200, 2000 or 3000. *Corwin Software, 10066 West Mawman Avenue, Waukegan, IL 60087-2431, (312) 623-4114, $30.*

DPATH+Plus, a disk file redirection utility that allows your programs to read and/or write any file, in any directory, on any disk drive. Requires Tandy 1000, 1200 or 3000. *Personal Business Solutions, Inc., P.O. Box 757, Frederick, MD 21701, (301) 865-3376, $45.*

Fortune Teller, tells your fortune based on Hindu astrology. Requires Tandy 1000, 1200, 2000 or 3000. *XYTON, Brown-Wagh Publishing, Inc., 800 Charcot Avenue 110, San Jose, CA 95131, $39.95.*

Golf Classic and Battling Bugs, two games that teach angles and length estimation. Requires Tandy 1000, 1200 or 3000 and graphics option. *Milliken Publishing Co., EduFun! Division, 1100 Research Boulevard, St. Louis, MO 63132, (314) 991-4220, $34.95.*

Gulp!! and **Frenzy**, two math skill development programs where the student tries to save small fish from being eaten by larger fish or an alligator by correctly answering math problems. Requires Tandy 1000, 1200 or 3000 and graphics option. *Milliken Publishing Co., EduFun! Division, 1100 Research Boulevard, St. Louis, MO 63132, (314) 991-4220, $34.95.*

HomeWord Plus, a personal word processing package with spelling checker. Requires Tandy 1000, 1200 or 3000. *Sierra On-Line, Inc., Coarsegold, CA 93614, $69.95.*

The Islands of Beta, an educational Adventure game which teaches problem solving, logic and creative thinking. Requires Tandy 1000, 1200 or 3000 and graphics option. *Milliken Publishing Co., EduFun! Division, 1100 Research Boulevard, St. Louis, MO 63132, (314) 991-4220, $34.95.*

King's Quest II, an animated Adventure game. Journey with King Graham and help him unlock the secrets that may lead him to the enchanted maiden for whom he longs. Requires Tandy 1000, 1200 or 3000 and graphics option. *Sierra On-Line, Inc., avaiable through Radio Shack stores nationwide, $39.95.*

Mastering the Tandy 2000, a book by Dan Keen and Dave Dischert which goes beyond the material in the standard owner's manuals. Provides actual working program examples. *Tab Books, Blue Ridge Summit, PA 17214, $10.95.*

MathFun Math Sequences, a series of math-oriented educational software combining drill exercises with positive reinforcement. Available packages: Addition Sequences, Division Sequences and Subtraction Sequences. Requires Tandy 1000, 1200 or 3000 and graphics option. *Milliken Publishing Co., EduFun! Division, 1100 Research Boulevard, St. Louis, MO 63132, (314) 991-4220, $34.95 each.*

Portable Light, a battery-operated, adjustable light for use with portable computers. Helps you read portable computer LCD screens in low-light conditions. *AMRO Computer Services, P.O. Box 948, Lake Oswego, OR 97034, (503) 692-5926, $6.99.*

Rud's Review, an electronic guide to mutual funds. Requires Tandy 1000, 1200 or 3000. *Visual Information Systems, Inc., P.O. Box 42106, Washington, DC 20015, 1-800-638-2000, $39.*

Software Bridge, converts documents generated with one word processing package into documents compatible and editable with another word processing package. Supports *DisplayWrite3*, *MultiMate*, *Samna Word III*, *Word MARC*, *WordPerfect* and *WordStar*. Requires Tandy 1000, 1200 or 3000, 512K. *Systems Compatibility Corporation, One East Wacker Drive, Suite 1320, Chicago, IL 60601, (312)329-0700, $249.*

By awarding a *Seal*, the magazine certifies the program *does exist*, but this *does not* constitute any guarantee of satisfaction. As soon as possible, these hardware or software items will be forwarded to PCM's reviewers for evaluation.

Using *BAREAD 2.1*

Bar code listings must be read in numerical order beginning with Line 1 and continuing through the last line of the listing. The computer display is used to prompt you as to which line to scan and give you warning messages should you happen to get out of step.

When you run *BAREAD*, it asks you to scan the first line of the bar code listing. This line contains the name of the program as well as the beginning of the program itself. The computer will sound a high-pitched beep whenever it's ready for you to scan a line. After a line has been successfully read, you'll hear a lower beep. A "blip-bloop" sound prompts you to turn your attention to the screen for a message. You'll hear this when you accidentally scan a line out of sequence.

After reading the first line, you continue scanning with the second line. Remember to wait for a high beep before scanning and then listen for a low beep to indicate a successful read.

Once the last line of the listing has been scanned, *BAREAD* will return control to the Tandy 100/200 menu screen. Note that the program you just scanned is now in the directory with a .DO extension.

The final step is to convert the .DO text file to a normal BASIC program. This is done quite simply by going to BASIC and loading the file with a command such as LOAD"TEST .DO" (if the program name were TEST). The program will load into BASIC and will be ready to run. To save the program in BASIC's compressed format (.BA extension), you'd type SAVE"TEST" (if the program were named TEST). You may then kill the .DO file with KILL "TEST.DO".

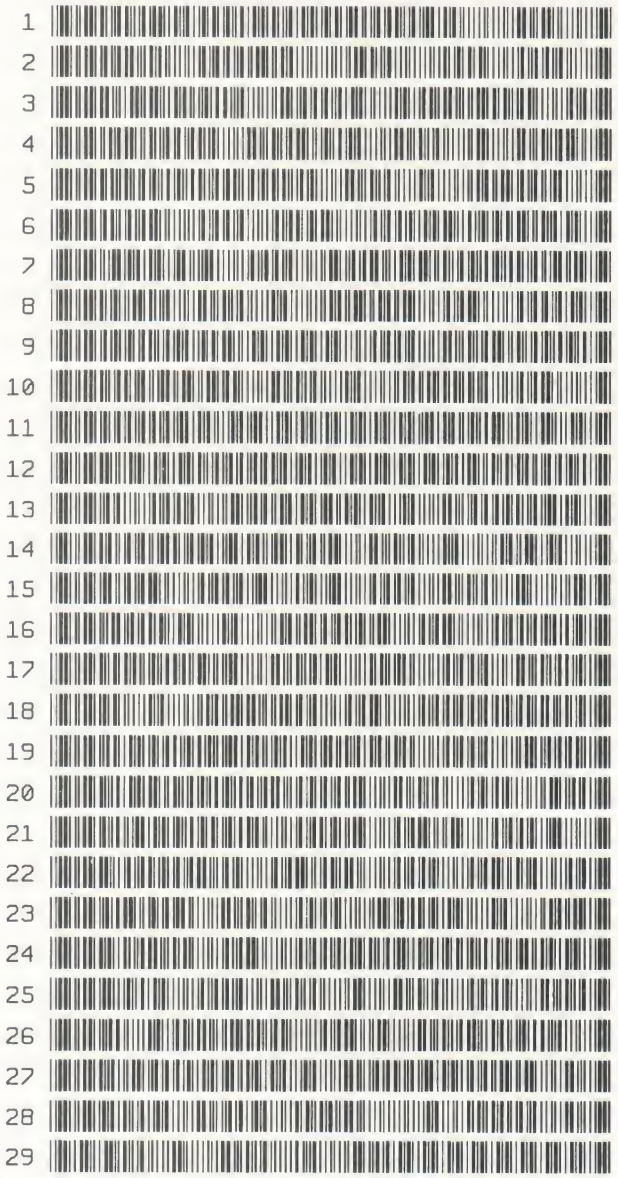
BAREAD 2.1

```
1ØØØ ' *** Initialize ***
1Ø1Ø ON ERROR GOTO 1Ø4Ø
1Ø2Ø CLEAR 1ØØØ:MAXFILES=2
1Ø3Ø GOTO 1Ø5Ø
1Ø4Ø IF ERR=5 THEN RESUME NEXT
1Ø5Ø ON ERROR GOTO Ø
1Ø6Ø RUNM "B3OF9"
1Ø7Ø OPEN "WAND:" FOR INPUT AS #1
1Ø8Ø UC%=-1
1Ø9Ø PC$="Ø123456789ABCDEFGHIJKLMNOPQRST
UVWXYZabcdefghijklmnopqrstuvwxyz- $+"
11ØØ DIM RW$(36)
111Ø ER$(1)="You must scan line 1 first!
"
112Ø ER$(2)="You've SKIPPED a line!"
113Ø ER$(3)="You've ALREADY SCANNED this
 line!"
114Ø ER$(4)="Code not PCM2/39 format!"
115Ø ER$(5)="Command not applicable here
!"
116Ø ER$(6)="You cannot skip this line!"
117Ø ER$(7)="Selected resume file not in
 computer!"
118Ø ' *** Read Reserved Words List ***
119Ø DATA BEEP,CLEAR,CLOSE,DATA,DEFDBL,D
EFINT,DEFNG,DEFSTR,ELSE,GOSUB,GOTO
12ØØ DATA INKEY$,INPUT,INSTR(,LCOPY,LEFT
$(,LINE(,LOADM,LPRINT,USING,MAXFILES
121Ø DATA MID$(,NEXT,PEEK,POKE,POWER,PRE
SET(,PRINT,READ,RESTORE,RETURN,RIGHT$(
122Ø DATA SOUND,SPACE$(,STRING$(,THEN
123Ø FOR I%=1 TO 36:READ RW$(I%):NEXT I%
124Ø ' *** Procedure Begins Here ***
125Ø CLS:PRINT@44,"PCM Bar Code Program
Reader v2.1"
126Ø LINE(2Ø,4)-(219,18),1,B:LINE(22,6)-
(217,16),1,B
```

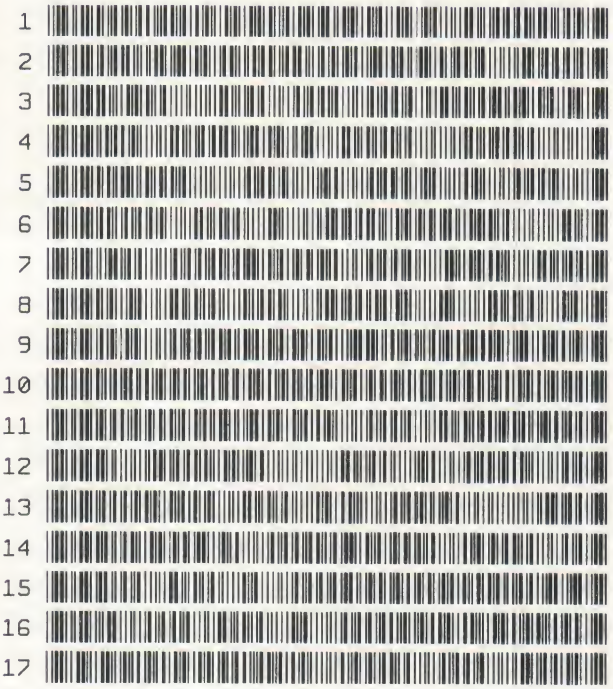
```
127Ø NN%=1
128Ø GOSUB 166Ø:IF ER%>Ø THEN GOSUB 162Ø
:GOTO 128Ø
129Ø IF LL%=Ø AND INSTR("YN",IL$)>Ø THEN
 ER%=5:GOSUB 162Ø:GOTO 128Ø
13ØØ IF LL%=Ø THEN ON INSTR("ALSR",IL$)
GOTO 182Ø,189Ø,198Ø,2Ø5Ø
131Ø IF LL%=1295 THEN 135Ø
132Ø IF LL%<>NN% AND NN%=1 THEN ER%=1:GO
SUB 162Ø:GOTO 128Ø
133Ø IF LL%<NN% THEN ER%=3:GOSUB 162Ø:GO
TO 128Ø
134Ø IF LL%>NN% AND NN%>1 THEN ER%=2:GOS
UB 162Ø:GOTO 128Ø
135Ø IL$=RIGHT$(IL$,19)
136Ø IF LL%=1 AND NN%>Ø THEN GOSUB 178Ø
137Ø CL$=CL$+IL$
138Ø FOR I%=1 TO LEN(CL$)
139Ø    CH$=MID$(CL$,I%,1)
14ØØ    IF CH$="%" THEN GOSUB 151Ø:IF NL
% THEN 147Ø ELSE GOTO 144Ø
141Ø    IF CH$="/" THEN GOSUB 155Ø:IF NL
% THEN 147Ø ELSE GOTO 144Ø
142Ø    IF CH$="." THEN UC%=NOT(UC%):GOT
O 145Ø
143Ø    IF CH$=>"A" AND CH$<="Z" AND NOT
(UC%) THEN CH$=CHR$(ASC(CH$)+32)
144Ø    XX$=XX$+CH$:IF RIGHT$(XX$,1)=CHR
$(13) THEN PRINT#2,XX$;:XX$="":UC%=-1
145Ø NEXT I%
146Ø CL$=""
147Ø PRINT@2ØØ,SPACE$(8Ø);
148Ø IF LL%<>1295 THEN NN%=LL%+1:GOTO 12
8Ø
149Ø ' *** Done ***
15ØØ CLOSE:CALL 618Ø7!:CLEAR 5ØØ,HIMEM:M
ENU
151Ø ' *** Decode Reserved Word ***
152Ø NL%=Ø:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="%":GOTO 154Ø
153Ø I%=I%+1:CH$=RW$(INSTR(PC$,MID$(CL$,
I%,1)))
154Ø RETURN
155Ø ' *** Decode Hex and Control Charac
ters ***
156Ø NL%=Ø:IF I%>LEN(CL$)-1 THEN NL%=-1:
CL$="/":GOTO 161Ø
157Ø I%=I%+1:IF INSTR("/%.",MID$(CL$,I%,
1))>Ø THEN CH$=MID$(CL$,I%,1):GOTO 161Ø
158Ø IF I%>LEN(CL$)-1 THEN NL%=-1:CL$=RI
GHT$(CL$,2):GOTO 161Ø
159Ø HX$=MID$(CL$,I%,2):CH$=CHR$((INSTR(
"Ø123456789ABCDEF",LEFT$(HX$,1))-1)*16+I
NSTR("Ø123456789ABCDEF",RIGHT$(HX$,1))-1
)
16ØØ I%=I%+1
161Ø RETURN
162Ø ' *** Error Codes ***
163Ø SOUND 5ØØØ,1Ø:SOUND 8ØØØ,1Ø:SOUND 5
ØØØ,1Ø
164Ø PRINT@22Ø-.5*LEN(ER$(ER%)),ER$(ER%)
;
165Ø RETURN
166Ø ' *** Get Code Line ***
167Ø PRINT@173,"";:PRINT USING "Scan lin
e ###";NN%
168Ø IF NN%=-1 THEN PRINT@173,"Scan any
line":GOTO 17ØØ
169Ø SOUND 5ØØ,5
17ØØ INPUT#1,IL$:ER%=Ø
171Ø FOR I%=1 TO LEN(IL$)
172Ø IF MID$(IL$,I%,1)="!" THEN MID$(IL$
,I%,1)="."
173Ø NEXT I%
174Ø IF LEN(IL$)<>1 AND LEN(IL$)<>21 THE
N ER%=4:RETURN
175Ø IF LEN(IL$)=1 THEN LL%=Ø:RETURN
176Ø LL$=LEFT$(IL$,2):LL%=(INSTR("Ø12345
6789ABCDEFGHIJKLMNOPQRSTUVWXYZ",LEFT$(LL
$,1))-1)*36+INSTR("Ø123456789ABCDEFGHIJK
LMNOPQRSTUVWXYZ",RIGHT$(LL$,1))-1
177Ø RETURN
178Ø ' *** Open Program File ***
179Ø PN$=LEFT$(IL$,6):IL$=RIGHT$(IL$,LEN
(IL$)-6)
18ØØ OPEN PN$ FOR OUTPUT AS #2
181Ø RETURN
182Ø ' *** Abort ***
183Ø BEEP:BEEP:BEEP
184Ø PRINT@2Ø9,"ABORT!  Are you sure?";
185Ø INPUT#1,AN$
186Ø IF INSTR("YN",AN$)=Ø THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 185Ø
187Ø PRINT@2ØØ,SPACE$(8Ø);
188Ø IF AN$="Y" THEN CLOSE:KILL PN$+".DO
":GOTO 149Ø ELSE GOTO 128Ø
189Ø ' *** Skip Line ***
19ØØ IF NN%=1 THEN ER%=6:GOSUB 162Ø:GOTO
 128Ø
191Ø BEEP:BEEP:BEEP
192Ø PRINT@21Ø,"SKIP!  Are you sure?"
193Ø INPUT#1,AN$
194Ø IF INSTR("YN",AN$)=Ø THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 193Ø
195Ø PRINT@2ØØ,SPACE$(8Ø);
196Ø IF AN$="Y" THEN NN%=NN%+1
197Ø GOTO 128Ø
198Ø ' *** Stop & Save ***
199Ø BEEP:BEEP:BEEP
2ØØØ PRINT@2Ø7,"STOP & SAVE! Are you sur
e?";
2Ø1Ø INPUT#1,AN$
2Ø2Ø IF INSTR("YN",AN$)=Ø THEN BEEP:PRIN
T@251,"Scan 'YES' or 'NO'":GOTO 2Ø1Ø
2Ø3Ø PRINT@2ØØ,SPACE$(8Ø);
2Ø4Ø IF AN$="Y" THEN 149Ø ELSE GOTO 128Ø
2Ø5Ø ' *** Resume ***
2Ø6Ø IF NN%<>1 THEN ER%=5:GOSUB 162Ø:GOT
```

```
0 128Ø
2Ø7Ø PRINT@254,"Resume Mode";
2Ø8Ø NN%=1:GOSUB 166Ø
2Ø9Ø IF LL%=Ø THEN ER%=5 ELSE IF LL%<>1
THEN ER%=1
21ØØ IF ER%>Ø THEN GOSUB 162Ø:GOTO 2Ø6Ø
211Ø PN$=MID$(IL$,3,6)
212Ø ON ERROR GOTO 214Ø
213Ø OPEN PN$ FOR INPUT AS #2:GOTO 217Ø
214Ø RESUME 215Ø
215Ø CLOSE #2
216Ø ER%=7:GOSUB 162Ø:GOTO 127Ø
217Ø CLOSE #2:OPEN PN$ FOR APPEND AS #2
218Ø NN%=-1:GOTO 128Ø
```

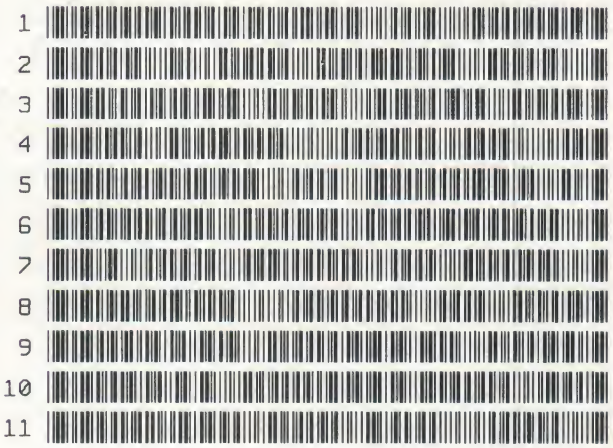
APRIL.BA (FROM PAGE 102)

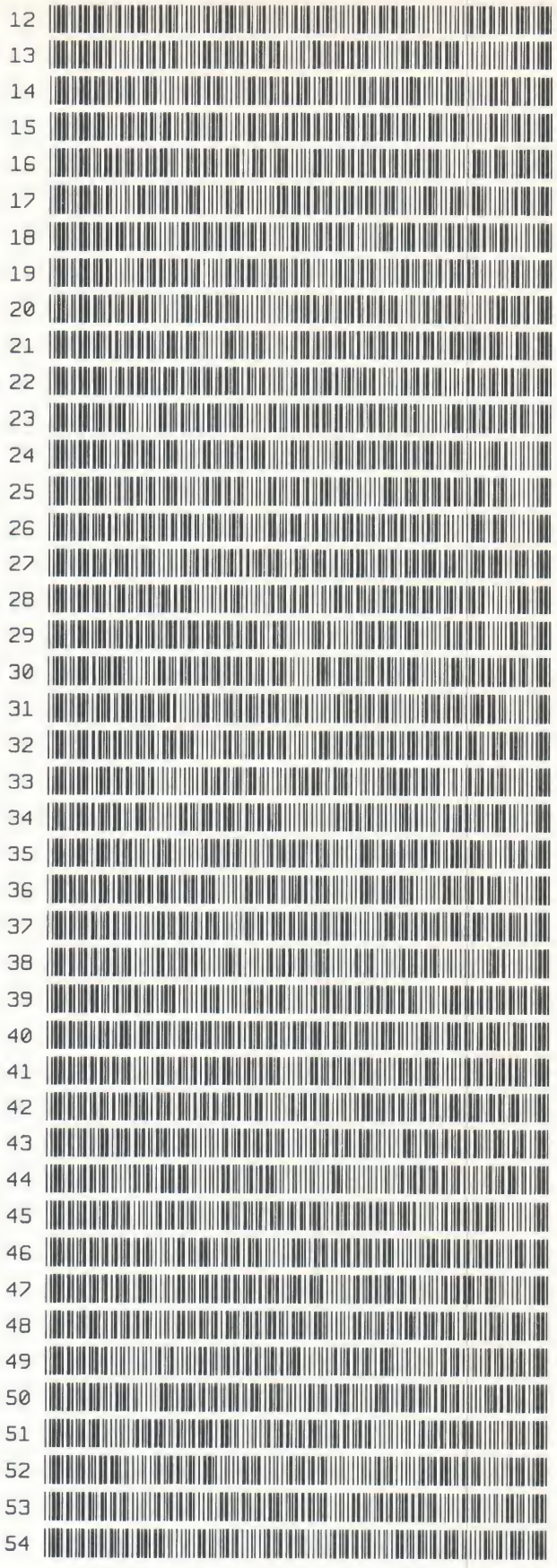


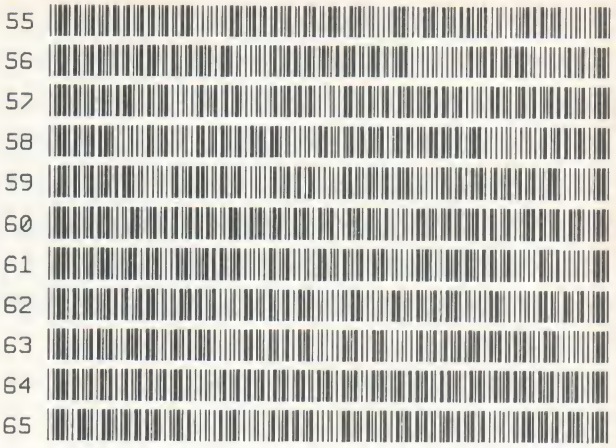
BURY.BA (FROM PAGE 102)



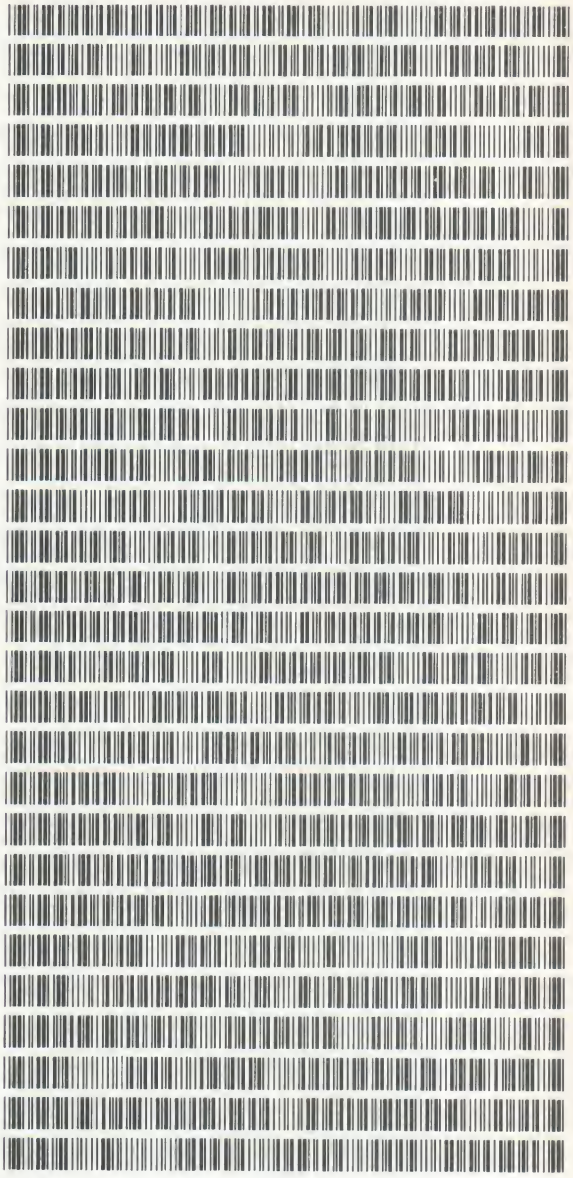
BOA1.BA (FROM PAGE 38)

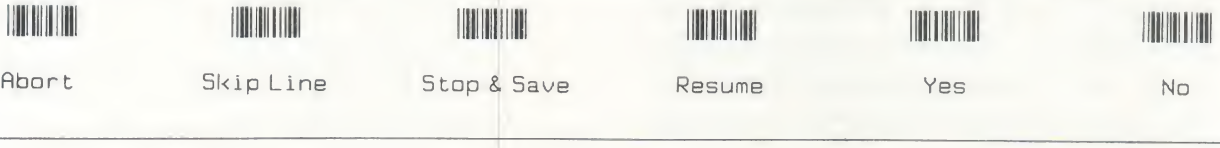


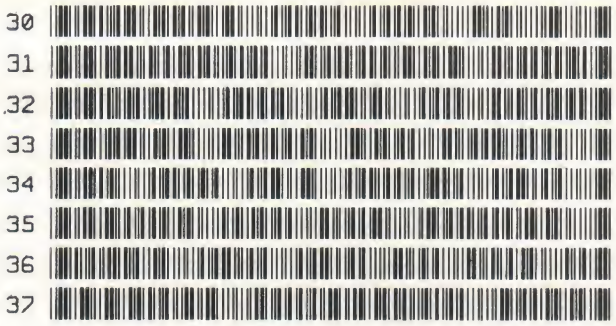




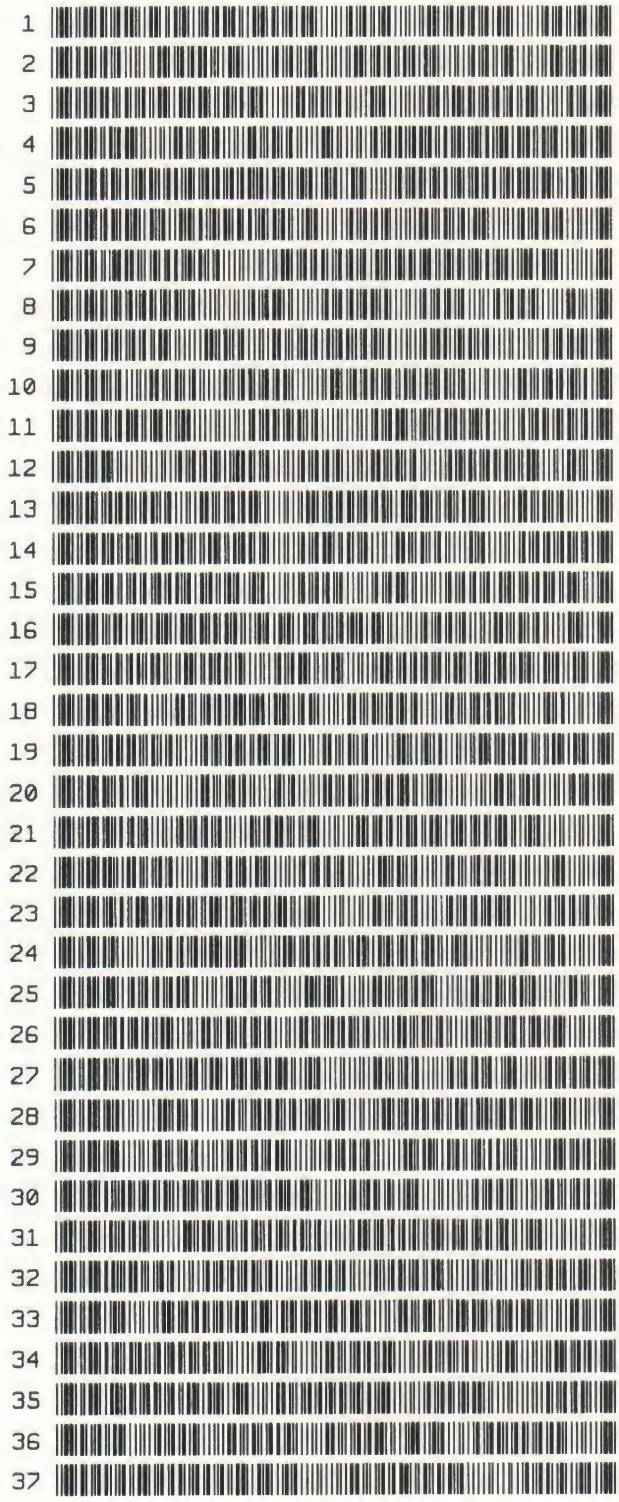
BOA2.BA (FROM PAGE 39)

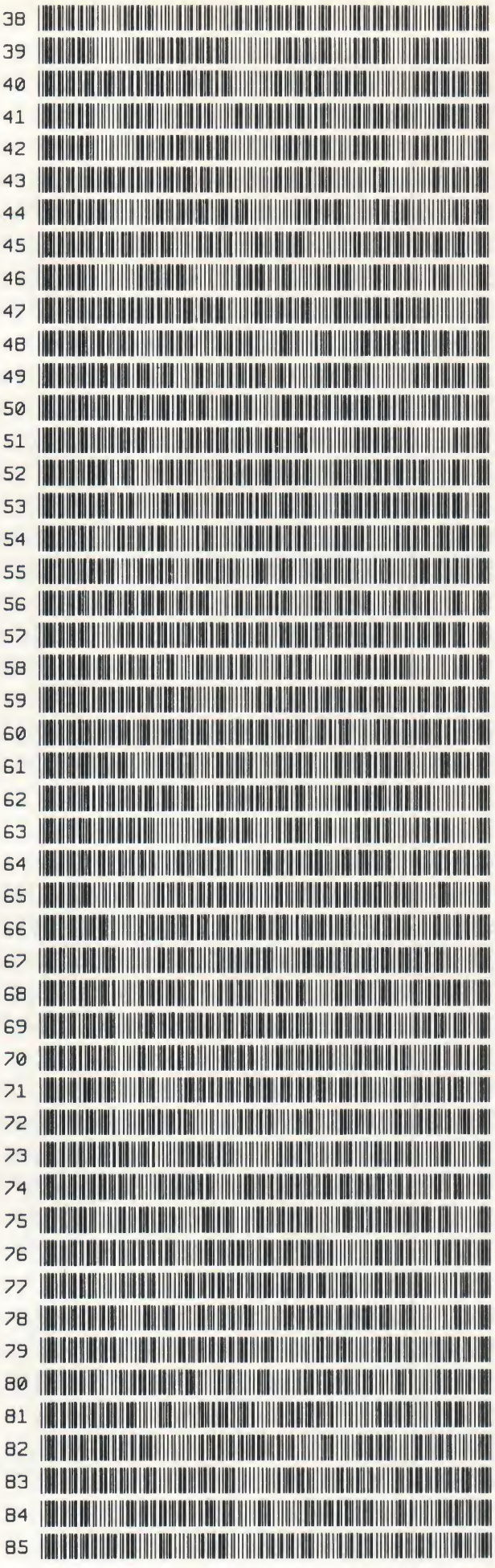




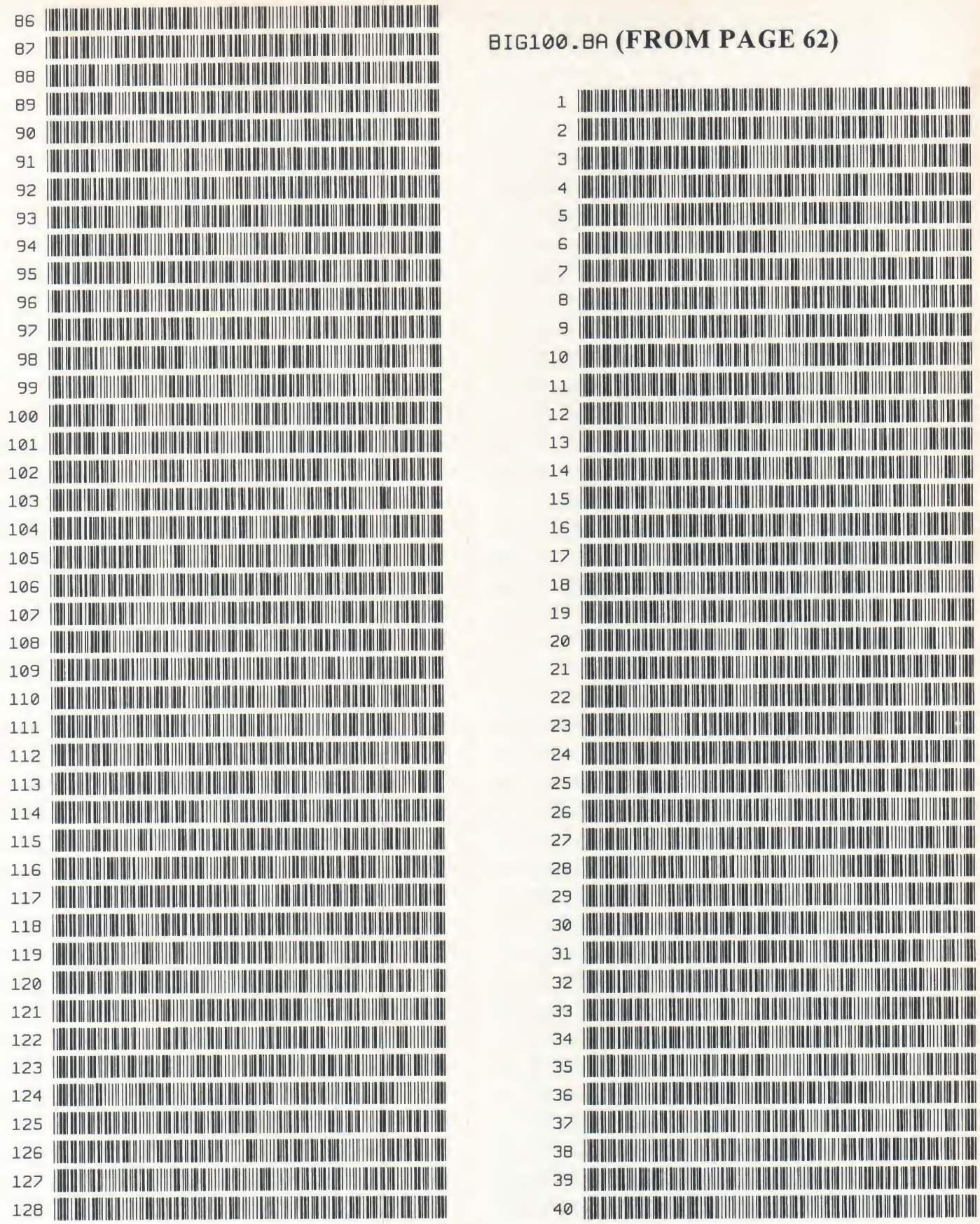


MCODE.BA (FROM PAGE 24)





BIG100.BA (FROM PAGE 62)

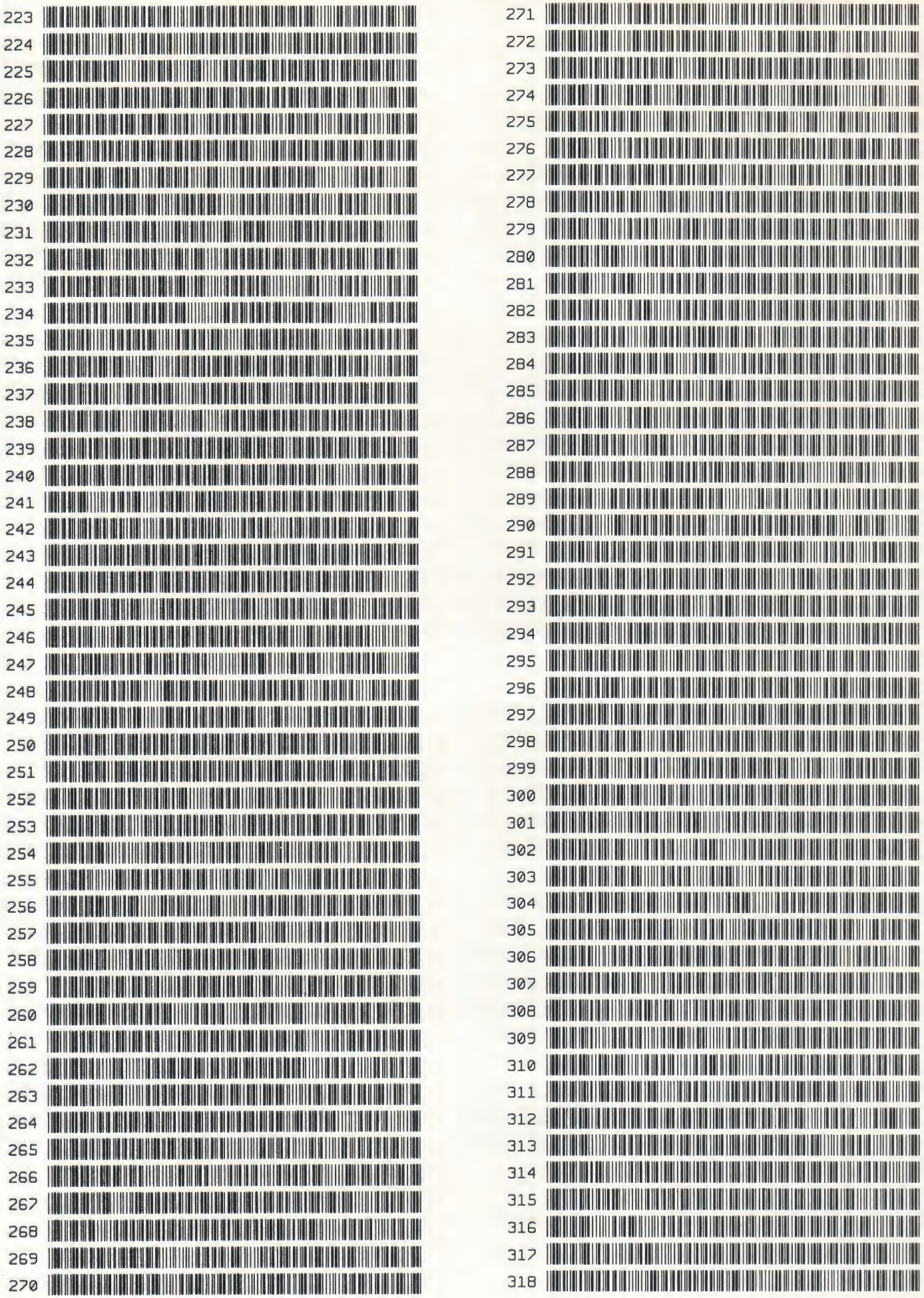


Abort Skip Line Stop & Save Resume Yes No

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222

Abort	Skip Line	Stop & Save	Resume	Yes	No



Abort	Skip Line	Stop & Save	Resume	Yes	No

The Rackseller

ALABAMA
Birmingham — Jefferson News Co.
Florence — Anderson News Co.
Greenville — M&B Electronics/Radio Shack
Madison — Madison Books

ARKANSAS
Fayetville — Vaughn Electronics/Radio Shack

ARIZONA
Phoenix — Computer Pro
Tri-Tek
Tucson — Anderson News Co.

CALIFORNIA
Citrus Heights — Software Plus
Elk Grove — Micro Mainframe
Half Moon Bay — Strawflower Electronics
Livermore — Software Galeria
North Hollywood — Levity Distributors
Santa Rosa — Sawyer's News
Sunnyvale — Computer Literacy

COLORADO
Salida — Radio Shack

DELAWARE
Middletown — Delmar Co.
Wilmington — Normar Inc.—The Smoke Shop

FLORIDA
Boca Raton — Software, Software, Inc.
Davie — Software Plus More
Ft. Lauderdale — Software Connection
N. Miami Beach — Almar Bookstore
Orlando — Book Mania
Panama City — Computer Systems Group
Pensacola — Anderson News Co.
Tallahassee — Anderson News Co.

GEORGIA
Atlanta — Guild News Agency

ILLINOIS
Chicago — Chas. Levy Distr.
Kroch's & Brentano's Jackson Street
Kroch's & Brentano's Wabash Ave.
Prairie News Agency
Oakbrook — Kroch's & Brentano's

INDIANA
Highland — Computer Health Consultants
Scottsburg — Radio Shack of Scottsburg

IOWA
Ames — Myers TV & Stereo Repair

KANSAS
Wichita — Lloyd's Radio

KENTUCKY
Georgetown — Goodwin Electronics
Louisville — Hawley-Cooke Bookstores
Prospect — Falsoft, Inc.

LOUISIANA
Gretna — Computer Supply Store
Slidell — Radio Shack

MASSACHUSETTS
Littleton — Computer Plus
Marlboro — Radio Shack

MICHIGAN
Rochester — Rochester Book Center
Sterling Heights — Software City
Programs, Inc.
Wyoming — Gerry's Book Co.

MISSOURI
Independence — Digital Enterprises, Inc.
St. Louis — Software Center

NEBRASKA
Lincoln — Hobby Town

NEW HAMPSHIRE
West Lebanon — Verham News Corp.

NEW JERSEY
Clinton — Micro World II
Hackensack — Total Circulation Service
Pennsville — Dave's Electronic Radio Shack

NEW MEXICO
Alamogardo — New Horizons Computer Systems
Albuquerque — News and Bookstore
Page One Newsstand

NEW YORK
Fredonia — On Line: Computer Access Center
Hudson Falls — G.A. West & Co.

NORTH CAROLINA
Cary — News Center in Cary Village
Sylva — Total Data Systems

NORTH DAKOTA
Fargo — Computer Associates

OHIO
Blanchester — J R Computer Control
Cincinnati — Cinsoft
Kenton — T.W. Hogan & Assoc., Inc.
Mount Orab — Mount Orab Radio Shack
Toledo — Leo's

OKLAHOMA
Oklahoma City — Merit Computers

OREGON
Portland — Fifth Avenue News
Woodburn — Pioneer Electronics

PENNSYLVANIA
Philadelphia — Newsy
Phoenixville — Stevens' Radio Shack
Wind Gap — Micro World

RHODE ISLAND
Newport — Kelly's Variety

TENNESSEE
Knoxville — Anderson News Co.
Memphis — Computer Center
Nashville — Mosko's Book Store
Smyrna — Delker Electronics, Inc.

TEXAS
Bay City — Impulse Computers
Ft. Worth — A & A International

VIRGINIA
Alexandria — Alonso Book & Period
Norfolk — I-O Computer
Wytheville — Wytheville Electronics

WASHINGTON
Kennewick — Satellite TV Specialists
Mt. Lake Terrace — Emerald Computer

WEST VIRGINIA
Lesage — Pioneer Technology

WISCONSIN
Ladysmith — Electronics, Etc.
Madison — 20th Century Books

ARGENTINA
Cordoba — Informatica Y Telecomunicaciones

CANADA:
ALBERTA
Calgary — Rainbow Software Services
Edmonton — CMD Micro
Kelly Software Distributors
BRITISH COLUMBIA
Sidney — Sidney Electronics
QUEBEC
LaSalle — Messageries De Presse Benjamin
NOVA SCOTIA
Halifax — Atlantic News

Also available at selected B.Dalton Bookseller stores

Advertisers Index

PCM Advertising and Marketing Representatives

Cindy Shackleford
Director, West Coast Office
12110 Meridian South — Suite 8
P.O. Box 73-578
Puyallup, WA 98373-0578
(206) 848-7766

Kim Vincent
Advertising Representative
The Falsoft Building
P.O. Box 385
Prospect, KY 40059
(502) 228-4492

PCMfest

Chicago

May 23-25

RAINBOWfest

Your admission to PCMfest also entitles you to attend RAINBOWfest, the highly popular show for the Tandy Color Computer which will run concurrently with PCMfest at the same location.

You're invited to PCMfest, a great show that's just for your Tandy computer. Sponsored by PCM, *The Personal Computer Magazine for Tandy Computer Users*, it's a wonderful way to meet and exchange information with those who share your interest in the new generation of Tandy computers.

You'll also discover the greatest variety of products ever for your computer because all of the favorites and a lot of new ones will be featured in the exhibits of PCM advertisers. Try out that new program and take it home that very day!

Along with other PCM readers, you'll meet the top national experts on your computer, including those who write for or who are written about in PCM. They will answer your questions on the spot.

PCMfest will also include a comprehensive lineup of free seminars on topics of immediate concern — and all of them designed to help you get the most out of your Tandy computer.

The Hyatt Regency Woodfield will be offering special rates ($60, single or double room) for PCMfest. The show opens Friday evening with a 7 p.m. to 10 p.m. session. It's a daytime-only show Saturday — the exhibits open at 10 a.m. and run continuously until 6 p.m. On Sunday, the exhibit hall opens at 11 a.m. and closes at 4 p.m.

Tickets may be obtained directly from PCM. We'll also send you a special reservation form so you can take advantage of the special room rate. Come to PCMfest and let's celebrate the new generation of Tandy computers!

YES, I'm coming to Chicago! I want to save by buying tickets now at the special advance sale price.

Please send me:

_____ three-day tickets at $9 each total _____

_____ one-day tickets at $7 each total _____

(Circle one) Friday Saturday Sunday

_____ Saturday Breakfast at $12 each total _____

Handling Charge $1 _____

TOTAL ENCLOSED _____

(U.S. Currency Only, Please)

☐ Also send me a hotel reservation card for the Hyatt Regency Woodfield ($60, single or double room).

Name _____

Address _____

City _____ State _____

Telephone _____ ZIP _____

Company _____

☐ Payment enclosed, OR ☐ Charge to:

☐ VISA ☐ MasterCard ☐ American Express

Account Number _____

Exp. Date _____

Signature _____

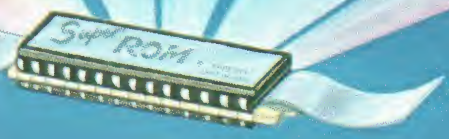
Make checks payable to: PCM. Mail to: PCMfest, The Falsoft Building, P.O. Box 385, U.S. Hwy. 42, Prospect, KY 40059. To make reservations by phone, call: (502) 228-4492.

Advance ticket deadline: May 16, 1986. Orders received less than two weeks prior to show opening will be held for you at the door. Tickets will also be available at the door at a slightly higher price. Children under 4, free; 4 and over, full admission price.

SUPER ROM

Lucid Spreadsheet **Write ROM** **Database** **Outliner**

"The Tandy Portable Drive with one button boot!"



NOW AVAILABLE ON THE M-200

NOW YOU CAN REALLY HAVE IT ALL!

All on one ROM. Truly the finest four programs available for the Model 100 — guaranteed. Try it for 30 days. If you aren't blown away by the excellence return it for a full refund.

$199.95

The four best programs for the Model 100 all on one ROM. 32K of power without using any RAM for program storage. This is the PCSG Snap-In ROM that just presses easily into the little ROM socket in the compartment on the back. You access the four right from the main menu like built-ins.

Write ROM — the definitive word processor for the Model 100. Function key formatting or dot commands. Search and replace. Library feature — inserts words, phrases or whole documents into text from just a code. MAP lets you see a picture of your document. In all there are 60 features and functions. No one can claim faster operation. FORM lets you create interactive forms with on-screen prompts that you can answer from the keyboard. Nothing else for the Model 100 compares with the features of Write ROM. Exactly the same as the Write ROM sold as a single program. Infoworld says it "makes the Model 100 a viable writing unit . . . surpassed our highest expectations for quality and clarity."

Lucid Spreadsheet: This is the one PICO magazine says "blows Multiplan right out of the socket" and Infoworld performance rated as "excellent" and said "makes the Model 100 compute." Gives you features you cannot get with Lotus 123. Lets you build spreadsheets in your Model 100 that would consume 140-150K on a desktop. Program generating capability with no programming knowledge required. Variable column widths. Includes find and sort with function key control. It's fast, recalculates like lightning. No feature has been taken from the original, only new ones added.

Database: This is a relational data base like no other. You can do everything from mailing lists to invoices. No complicated pseudo-coding, you create input screens as simply as typing into TEXT. You are not limited by size; you can have as large an input screen as you wish. Prints out reports or forms, getting information from as many files as you like. Complete math between fields. Total interface with Lucid worksheets.

Outliner: Does everything that Thinktank does on a PC but a whole lot better. Includes a Sort for your headlines. Lets you have headlines of up to 240 characters. Has cloning, hoisting and sideways scroll up to 250 characters. Like Lucid, this one sets a new standard for outliners. This is the way to plan and organize your projects.

Present Lucid and Write ROM owners can upgrade for **$125**. If you have both it's **$100**.

As usual PCSG sells the Super ROM on a thirty day guarantee. If for any reason you are not satisfied, simply return it for a full refund.

We are excited about this product. Super ROM gives the Model 100 the true power of a desktop. No other multi-program ROM has software that compares. But don't take our word for it. We invite you to make that comparison yourself. Priced at $199.95 on Snap-In ROM.

Got stuck with somebody else's multi-ROM? We'll upgrade it for $100.

(214) 351-0564

PORTABLE COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd., #206, Dallas, TX 75229

MC, Visa, American Express, Check, or C.O.D.